

AD-A118 920

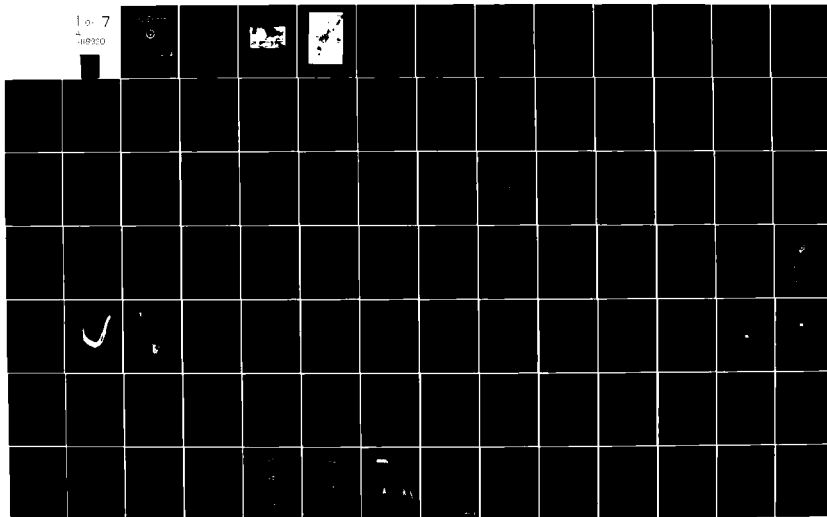
ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82

F/O 12/1

UNCLASSIFIED ARO-82-3

NL

1 of 7
18920



AD A118920

12

ARO Report 82-3 PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS CONFERENCE



Approved for public release; distribution unlimited. The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DTIC
SEP 3 1982
A

SPONSORED BY
THE ARMY MATHEMATICS STEERING COMMITTEE ON BEHALF OF

THE OFFICE OF
THE CHIEF OF RESEARCH, DEVELOPMENT AND
ACQUISITION

DTIC FILE COPY

82 08 3

U. S. Army Research Office

Report No. 82-3

August 1982

PROCEEDINGS OF THE 1982 ARMY NUMERICAL
ANALYSIS AND COMPUTERS CONFERENCE

Sponsored by the Army Mathematics Steering Committee

HOST

U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi
3-4 February 1982

Approved for public release; distribution unlimited.
The findings in this report are not to be construed
as an official Department of the Army position, un-
less so designated by other authorized documents.

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, North Carolina



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

FOREWORD

The 1982 Army Numerical Analysis and Computers Conference, sponsored by the Army Mathematics Steering Committee (AMSC), had as its host the U. S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, and was held on the dates 3-4 February 1982. A short "History" of and the "Mission" of the Waterways Experiment Station appeared in a booklet issued the attendees at this meeting. This information together with two photographs is reproduced next.



Administrative Headquarters

HISTORY

Following one of the Nation's great disasters—the 1927 flood on the Mississippi River—the Waterways Experiment Station was established in 1929 as a hydraulics laboratory to assist the Mississippi River Commission in developing and implementing comprehensive plans for flood control in the Lower Mississippi Valley. As the program advanced, it soon became necessary to establish a soils laboratory to aid in designing the levee system and ensure the adequacy of foundations. To support the military missions of the Corps of

Engineers during World War II, to meet the unique and challenging requirements of the postwar Space Age, and to keep abreast of the public's growing concern for protection and enhancement of our natural environment, functions and facilities were progressively added to produce capabilities in the many and diverse fields of endeavor described herein. Because of its nationwide scope of activity, the Waterways Experiment Station now operates under direct control of the Chief of Engineers.



Aerial view; administrative headquarters in left foreground

MISSION

The Waterways Experiment Station laboratory complex is the principal research, testing, and development facility of the U. S. Army Corps of Engineers. Its mission is to conceive, plan, and execute engineering investigations, and research and development studies, in support of the civil and military missions of the Chief of Engineers and other Federal agencies, through the operation of a complex of laboratories in the broad fields of hydraulics, soil and rock mechanics, concrete, expedient construction, nuclear and conventional weapons effects, nuclear and chemical explosives excavation, vehicle mobility, environmental relationships, engineering geology, pavements, protective structures, aquatic plants, water quality, and dredged material. On a reimbursable basis, the Waterways Experiment Station performs, on an extensive national scope, basic and applied research in these and related fields, develops methods and techniques, tests materials and equipment, and provides consulting services in its specialized fields of competence. Activities include model and prototype studies, engineering and analytical design studies including blast and shock effects, laboratory research concerning portland cement and bituminous concrete mixture proportioning, laboratory testing, field investigations, environmental and water-quality studies, dredged material research, technical advice and assistance on the use of nuclear explosives and large-yield chemical

explosives for excavation purposes, computer processing, analysis, programming, data preparation, graphics, and related service. Field investigation services include planning, accomplishment, and analysis of: comprehensive exploration of soil and rock formations; comprehensive examination and inspection of portland cement concrete structures in service including use of nondestructive testing procedures; instrumentation systems to measure water velocities and directions and to determine pressures, deflections, and strains in engineering structures; telemetry systems, principally for hydrologic data; and field inspection services including inspection of control laboratories and training of field personnel. Subject to approval by the Chief of Engineers, studies are also undertaken for other Defense and Federal agencies, private organizations, State Public Works, and foreign governments. The Waterways Experiment Station operates the Department of Defense Information Analysis Centers for Pavements and Soils Trafficability, Concrete Technology, Hydraulic Engineering, and Soil Mechanics. In connection with this fact-disseminating service, WES maintains an extensive scientific and engineering reference library and issues publications of general interest, which can aid materially in eliminating duplication of effort throughout the Corps of Engineers in the conduct of experimental studies.

* * * * *

The theme of the 1982 Conference was "Grid Techniques for Partial Differential Equations". Not only did all the invited speakers emphasize this important area but many of the authors of contributed papers treated it. Preceding the conference on the dates 1-2 February 1982 a tutorial entitled "Review of Finite Element/Finite Difference Methods for Partial Differential Equations" was offered by Professors S. Parter, C. de Boor, and J. Strikwerda of the Mathematics Research Center, University of Wisconsin, Madison, Wisconsin. The names of the invited speakers and the titles of their addresses are noted on the following page.

Speakers and Affiliation

Title of Address

Professor J. F. Thompson
Mississippi State University

ELLIPTIC GRID GENERATION TECHNIQUES

Dr. Patrick J. Roache
Ecodynamics Research Associates, Inc.

INTERACTIVE DESIGN OF LASER ELECTRODES
USING ELLIPTIC GRID GENERATION AND
SEMIDIRECT/MARCHING METHODS

Dr. James M. Hyman
Los Alamos Scientific Laboratory

THE STRUCTURED DESIGN OF ADAPTIVE MESH
METHODS FOR PDE'S

Dr. Dale A. Anderson
Iowa State University

SOLUTION ADAPTIVE GRIDS FOR PARTIAL
DIFFERENTIAL EQUATIONS

Those attending this meeting would like to take this occasion to express their appreciation to Mr. Marden B. Boyd, Chairman of Local Arrangements, for doing such an outstanding job of arranging physical accommodations and for handling the many problems they posed during the course of the conference.

Members of the AMSC would like to thank the speakers and all the other individuals who contributed to the success of this conference. They have asked that these proceedings be issued to enable those scientists that could not attend, as well as those present, to have a summary of the meeting.

TABLE OF CONTENTS*

<u>Title</u>	<u>Page</u>
Foreword	iii
Table of Contents	vii
Agenda	ix
Wescor - Boundary-Fitted Coordinate Code for General 2D Regions with Obstacles and Boundary Intrusions Joe F. Thompson	1
Flux-Corrected Transport in an Exponentially Stretched Grid Richard A. Schmalz, Jr.	77
Grid Generation Techniques for Projectile Configurations Charles J. Nietubicz, Karen R. Heavey and Joseph L. Steger	99
Application of Relative Coordinates in Hydrodynamics R. H. Multer	123
A Generalized Random Choice Method for Gas Dynamics James Glimm, Guillermo Marshall and Bradley Plohr	137
Formulation of Two-Phase Interior Ballistics Equations for Numerical Treatment Aivars K. R. Celmins and James A. Schmitt	149
Free Boundary Problems with Nonlinear Source Terms Gunter H. Meyer	245
Numerical Solution to an Autofrettaged Tube with Constraining Walls and End Closures Peter C. T. Chen	253
Finite Element Modeling of the Vulnerability of U. S. and Foreign Land Mines to Blast Loads Frederick H. Gregory and Aaron D. Gupta	267
Numerical Results of Transient Two-Dimensional Heat Conduction R. Yalamanchili	293
Salome, A Structured and Logically Minimal Ensemble of Programming Constructs Royce W. Soanes, Jr.	305

*This Table of Contents lists only the papers that are published in this Technical Manual. For a list of all the papers presented at the 1981 Army Numerical Analysis and Computers Conference see the copy of the Agenda.

A Finite Difference Program for Computing the Thermoelastic- Plastic Response of Lined Gun Barrels John D. Vasilakis	323
Accurate Computer Arithmetic for Scientific Computation L. B. Rall	343
Mathematical Software and Mathematical Software Libraries Alfred H. Morris, Jr.	357
ADI Procedures for Solving the Shallow-Water Equations in Transformed Coordinates H. L. Butler and Y. P. Sheng	365
Asymptotic and Numerical Methods for Vector Systems of Singularly-Perturbed Boundary Value Problems Joseph E. Flaherty and Robert E. O'Malley, Jr.	381
An Integral Equation for the Design of Magnetic Field Coils J. F. Schenck, M. A. Hussain, W. A. Edelstein and B. Noble	397
Element Type Comparison in Basin Oscillation Analysis Mark D. Prater and Keith W. Bedford	415
A Tsunami Generation and Propagation Model Driven by Vertical Seabed Movements Jeff Earickson	429
Theory and Calculation of the Non-Linear Energy Transfer Between Sea Waves in Deep Water Barbara A. Tracy and Donald T. Resio	457
Constrained and Unconstrained Variational Finite Element Formulation of Solutions to a Stress Wave Problem - A Numerical Comparison Julian J. Wu and C. N. Shen	477
Numerical Solutions Using Adjoint Variational Formulation to Stress Wave Problems C. N. Shen and J. J. Wu	499
Finite Difference Methods for the Stokes and Navier-Stokes Equations John C. Strikwerda	517
A Three-Dimensional Numerical Model of Coastal, Estuarine and Lake Currents Y. P. Sheng and H. L. Butler	531
Solution Adaptive Grids for Partial Differential Equations Dale A. Anderson	575
Interactive Design of Laser Electrodes Using Elliptic Grid Generation and Semidirect/Marching Methods Patrick J. Roache	593
Attendees	608

AGENDA FOR THE
1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS CONFERENCE
3-4 February 1982
Vicksburg, Mississippi

Wednesday
3 February 1982

0815-0830 REGISTRATION (Auditorium)

0830 0900 WELCOMING REMARKS - LTC John O. Evans, III, Deputy Commander and
Director, US Army Engineer Waterways
Experiment Station, Vicksburg, Mississippi

0900-1000 KEYNOTE ADDRESS (Auditorium)

CHAIRPERSON - Dr. B. Z. Jenkins, US Army Missile Command,
Redstone Arsenal, Alabama

SPEAKER - Professor J. F. Thompson, Mississippi State
University, Mississippi State, Mississippi

TITLE - ELLIPTIC GRID GENERATION TECHNIQUES

1000-1030 BREAK

1030-1225 TECHNICAL SESSION I (Main Conference Room)

CHAIRPERSON - Dr. Royce Soanes, Benet Weapons Laboratory,
Watervliet, New York

FLUX-CORRECTED TRANSPORT IN AN EXPONENTIALLY STRETCHED GRID

Dr. Richard A. Schmalz, Jr., US Army Engineer Waterways
Experiment Station, Vicksburg, Mississippi

GRID GENERATION TECHNIQUES FOR PROJECTILE CONFIGURATIONS

Drs. Charles J. Nietubicz, Karen R. Heavey and Joseph L. Steger,
Ballistic Research Laboratory, Aberdeen Proving Ground,
Maryland

MOVING FINITE ELEMENTS IN 2-D

Dr. Robert J. Gelinas, Science Applications, Inc., Pleasanton,
California

APPLICATION OF RELATIVE COORDINATES TO HYDRODYNAMIC PROBLEMS

Dr. Roger H. Multer, US Army Engineer Waterways Experiment
Station, Vicksburg, Mississippi

A MODIFIED UNIFORM SAMPLING METHOD FOR GAS FLOW IN A NOZZLE

Dr. Bradley Plohr, The Rockefeller University, New York, New York

1030-1225

TECHNICAL SESSION II (Auditorium)

**CHAIRPERSON - Dr. H. L. Butler, US Army Engineer Waterways
Experiment Station, Vicksburg, Mississippi**

**FORMULATION OF TWO-PHASE INTERIOR BALLISTICS EQUATIONS FOR
NUMERICAL TREATMENT**

**Drs. Aivars Celmins and James Schmitt, Ballistic Research
Laboratory, Aberdeen Proving Ground, Maryland**

MONOTONE METHODS FOR FREE BOUNDARY PROBLEMS

**Professor Gunter H. Meyer, Georgia Institute of Technology,
Atlanta, Georgia**

**NUMERICAL SOLUTION TO AN AUTOFRETTAGED TUBE WITH CONSTRAINING
WALLS AND END CLOSURES**

Dr. P. C. T. Chen, Benet Weapons Laboratory, Watervliet, New York

**FINITE ELEMENT MODELING OF THE VULNERABILITY OF U.S. AND FOREIGN
LAND MINES TO BLAST LOADS**

**Drs. Frederic H. Gregory and Aaron D. Gupta, Ballistic Research
Laboratory, Aberdeen Proving Ground, Maryland**

NUMERICAL RESULTS OF TRANSIENT TWO-DIMENSIONAL HEAT CONDUCTION

**Dr. R. Yalamanchili, U. S. Army Armament R&D Command, FC&SCWSL,
Dover, New Jersey**

1225-1330

LUNCH

1330-1430

GENERAL SESSION I (Auditorium)

**CHAIRPERSON - Dr. R. L. Launer, U. S. Army Research Office,
Research Triangle Park, North Carolina**

**INTERACTIVE DESIGN OF LASER ELECTRODES USING ELLIPTIC GRID
GENERATION AND SEMIDIRECT/MARCHING METHODS**

**Dr. Patrick J. Roache, Ecodynamics Research Associates, Inc.,
Albuquerque, New Mexico**

1430-1500 BREAK

1500-1700 TECHNICAL SESSION III (Main Conference Room)

CHAIRPERSON - Dr. D. S. Sodhi, US Army Cold Regions Research & Engineering Laboratory, Hanover, New Hampshire

APPLICATION OF THE PRINCIPAL COMPONENT METHOD TO TRAJECTORY ESTIMATION

Messrs. William S. Agee and Robert H. Turner, White Sands Missile Range, White Sands Missile Range, New Mexico

MULTIVARIATE B-SPLINES

Professor Carl de Boor, Mathematics Research Center, University of Wisconsin-Madison

SALOME, A STRUCTURED AND LOGICALLY MINIMAL ENSEMBLE OF PROGRAMMING CONSTRUCTS

Dr. Royce Soanes, Benet Weapons Laboratory, Watervliet, New York

A FINITE DIFFERENCE PROGRAM FOR COMPUTING THE THERMOELASTIC-PLASTIC RESPONSE OF LINED GUN BARRELS

Dr. John D. Vasilakis, Benet Weapons Laboratory, Watervliet, New York

ACCURATE COMPUTER ARITHMETIC FOR SCIENTIFIC COMPUTATION

Professor Louis B. Rall, Mathematics Research Center, University of Wisconsin-Madison

MATHEMATICAL SOFTWARE AND MATHEMATICAL SOFTWARE LIBRARIES

Dr. Alfred H. Morris, Jr., Naval Surface Weapons Center, Dahlgren, Virginia

1500-1640 TECHNICAL SESSION IV (Auditorium)

CHAIRPERSON - Dr. Aivars Celmins, Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

ADI PROCEDURES FOR SOLVING THE SHALLOW-WATER EQUATIONS IN TRANSFORMED COORDINATES

Dr. H. L. Butler, US Army Engineer Waterways Experiment Station, Vicksburg, Mississippi and Dr. Y. P. Sheng, Aeronautical Research Associates of Princeton, Inc., Princeton, New Jersey

ASYMPTOTIC AND NUMERICAL METHODS FOR VECTOR SYSTEMS OF
SINGULARLY PERTURBED BOUNDARY VALUE PROBLEMS

Dr. Joseph E. Flaherty, Rensselaer Polytechnic Institute, Troy,
New York, and Benet Weapons Laboratory, Watervliet, New York
and Dr. Robert E. O'Malley, Jr., Rensselaer Polytechnic
Institute, Troy, New York

BLOCK SPLITTING FOR THE CONJUGATE GRADIENT METHODS

Professors Seymour Parter, M. Steuerwalt and D. Kratzer,
Mathematics Research Center, University of Wisconsin-Madison

AN INTEGRAL EQUATION FOR MAGNET COIL DESIGN

Drs. J. F. Schenck, M. A. Hussain and W. A. Edelstein, General
Electric Company Corporate Research & Development, Schenectady,
New York, and Professor Ben Noble, Mathematics Research Center,
University of Wisconsin-Madison

TWO-DIMENSIONAL FINITE ELEMENT ANALYSIS OF FREE EDGE EFFECTS IN
LAMINATED COMPOSITES

Dr. Roshdy S. Barsoum, US Army Materiel Systems Analysis
Activity, Aberdeen Proving Ground, Maryland

Thursday
4 February 1982

0815-0915 GENERAL SESSION II (Auditorium)

CHAIRPERSON - Dr. N. Radakrishnan, US Army Engineer Waterways
Experiment Station, Vicksburg, Mississippi

THE STRUCTURED DESIGN OF ADAPTIVE MESH METHODS FOR PDE'S

Dr. James M. Hyman, Los Alamos Scientific Laboratory, Los
Alamos, New Mexico

0915-0945 BREAK

0945-1125 TECHNICAL SESSION V (Main Conference Room)

CHAIRPERSON - Mr. William S. Agee, White Sands Missile Range,
White Sands Missile Range, New Mexico

THEORETICAL AND EXPERIMENTAL BUCKLING LOADS OF FLOATING ICE
SHEETS

Dr. D. S. Sodhi, US Army Cold Regions Research & Engineering
Laboratory, Hanover, New Hampshire

ELEMENT TYPE COMPARISON IN BASIN OSCILLATION ANALYSIS

Dr. Mark D. Prater, US Army Engineer Waterways Experiment
Station, Vicksburg, Mississippi

A TSUNAMI GENERATION AND PROPAGATION MODEL DRIVEN BY VERTICAL
SEABED MOVEMENTS

Dr. Jeff A. Earickson, US Army Engineer Waterways Experiment
Station, Vicksburg, Mississippi

A HYBRID FINITE ELEMENT MODEL FOR WATER WAVES ARBITRARY
WAVELENGTH

Dr. James R. Houston, US Army Engineer Waterways Experiment
Station, Vicksburg, Mississippi

THEORY AND CALCULATION OF THE NONLINEAR ENERGY TRANSFER BETWEEN
SEA WAVES IN DEEP WATER

Drs. B. A. Tracey and D. T. Resio, US Army Engineer Waterways
Experiment Station, Vicksburg, Mississippi

0945-1125

TECHNICAL SESSION VI (Auditorium)

CHAIRPERSON - Mr. Henry Kahn, US Army Armament R&D Command,
Dover, New Jersey

CONSTRAINED AND UNCONSTRAINED VARIATIONAL FINITE ELEMENT
FORMULATION OF SOLUTIONS TO A STRESS WAVE PROBLEM - A NUMERICAL
COMPARISON

Drs. J. J. Wu and C. N. Shen, Benet Weapons Laboratory,
Watervliet, New York

NUMERICAL SOLUTIONS USING ADJOINT VARIATIONAL FORMULATION TO
STRESS WAVE PROBLEMS

Drs. C. N. Shen and J. J. Wu, Benet Weapons Laboratory,
Watervliet, New York

FINITE ELEMENT MODELING OF COMBINED SHEAR AND COMPRESSION WAVES
IN TUBES

Drs. Joseph F. Santiago and Bahaaeldin I. Shehata, Ballistic
Research Laboratory, Aberdeen Proving Ground, Maryland

FINITE DIFFERENCE SCHEMES FOR THE STOKES EQUATIONS

Professor John Strikwerda, Mathematics Research Center,
University of Wisconsin-Madison

ON EFFICIENT NUMERICAL SCHEMES FOR SOLVING 3-D NAVIER-STOKES
EQUATIONS

Dr. Y. Peter Sheng, Aeronautic. Research Associates of
Princeton, Inc., Princeton, New Jersey

1125-1225

GENERAL SESSION III (Auditorium)

CHAIRPERSON - Mr. Marden B. Boyd, US Army Engineer Waterways
Experiment Station, Vicksburg, Mississippi

SOLUTION ADAPTIVE GRIDS FOR PARTIAL DIFFERENTIAL EQUATIONS

Dr. Dale A. Anderson, Iowa State University, Ames, Iowa

1225

ADJOURN

WESCOR - BOUNDARY-FITTED COORDINATE CODE FOR
GENERAL 2D REGIONS WITH OBSTACLES AND BOUNDARY INTRUSIONS

Joe F. Thompson
Department of Aerospace Engineering
Mississippi State University
Drawer A
Mississippi State, MS 39762

Abstract

A code for the generation of boundary-fitted coordinate systems for general 2D regions with boundaries of arbitrary shape and with internal obstacles and boundary intrusions, arbitrary in shape and number, is described and instructions for input and use are given. The coordinate system is generated from the numerical solution of a system of elliptic partial differential equations with provision for controlling the spacing of the coordinate lines in the field. The transformed (computational) region is rectangular with the obstacles and intrusions transformed to slits and/or slabs. A small code to distribute points on various fundamental curves with exponential concentration is also described. This front-end code can be used to construct boundary point distributions for input to the coordinate code. A plot code for the coordinate system is also included. The boundary-fitted coordinate systems generated by this code may be used as a basis for the numerical solution of partial differential equations for any physical problem of interest.

Acknowledgement

The interest of Dr. Billy H. Johnson of the Waterways Experiment Station in this code and the many fruitful discussions with him during its development are gratefully acknowledged.

Prepared for Contract DACW39-78-C-0054
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

INTRODUCTION

The use of numerically-generated boundary-fitted curvilinear coordinate systems as the basis for numerical solution of partial differential equations on arbitrary regions is now well established. A comprehensive survey of the generation and use of these coordinate systems has recently appeared, Ref. [1], and the proceedings of a recent symposium devoted to this area, Ref. [2], cover the basic techniques involved, as well as applications in many areas.

Such coordinate systems have the property that some coordinate line is coincident with each segment of the boundary in the physical region, so that the complication of boundary shape is effectively removed from the problem. In the past decade the numerical generation of curvilinear coordinate systems has provided the key to the development of finite difference solutions of partial differential equations on regions with arbitrarily shaped boundaries. Although much of the impetus for these developments has come from fluid dynamics, the techniques are equally applicable to heat transfer, electromagnetics, structures, and all other areas involving field solutions.

With coordinate systems generated to maintain coordinate lines (surfaces in 3D) coincident with the boundaries, finite difference codes can be written which are applicable to general configurations without the need of special procedures at the boundaries. Even when the boundaries are in motion, the use of such coordinate systems allows all computation to be done on a fixed grid with a uniform square mesh in the transformed plane. This greatly simplifies the coding, particularly with regard to boundary conditions, which can now be represented without

need of interpolation. It is also possible to distribute the curvilinear coordinate lines in the physical plane with concentration of lines in regions of high gradients while maintaining the square grid in the transformed (computational) plane.

With such systems, the grid points may be thought of as a finite set of observers of the physical solution, stationed so as to be most effective in covering all of the action on the field. The structure of an intersecting net of families of coordinate lines allows the observers to be readily identified in relation to each other. This results in much more simple coding than would the use of a triangular structure or a random distribution of points. The grid generation system provides some influence of each observer on the others so that when one moves to get into a better position, its neighbors will follow in order to maintain smooth coverage of the field. The curvilinear coordinate system thus should cover the field, with coordinate lines (surfaces) coincident with all boundaries. The distribution of lines should be smooth, with concentration in regions of high gradient.

Numerical solutions of partial differential equations are done on the curvilinear coordinate system by first transforming all partial derivatives (or integrals) analytically so that the curvilinear coordinates, rather than the physical coordinates, become the independent variables. Normal and tangential derivatives at boundaries are similarly transformed. (These transformation relations are given in Ref. [3].) The result is a set of partial differential equations and boundary conditions in which all derivatives (and integrals) are with respect to the curvilinear coordinates. These equations may then be expressed as difference equations

on the square grid that is inherent in the transformed plane. There is thus no need for interpolation regardless of the shape of the boundaries or the distribution of the curvilinear coordinate lines in the field.

The present report concerns a code for the generation of boundary-fitted coordinate systems for general 2D regions with boundaries of arbitrary shape and with internal obstacles and boundary intrusions, arbitrary in shape and number. The code is described and instructions for input and use are given. Examples of the application of this code are given in Ref. [4]-[6]. The coordinate system is generated from the numerical solution of a system of elliptic partial differential equations with provision for controlling the spacing of the coordinate lines in the field. The transformed (computational) region is rectangular with the obstacles and intrusions transformed to slits and/or slabs. (This type of transformed configuration and its use are discussed in Ref. [3].) A small code to distribute points on various fundamental curves with exponential concentration is also described. This front-end code can be used to construct boundary point distributions for input to the coordinate code. A plot code for the coordinate system is also included. The boundary-fitted coordinate systems generated by this code may be used as a basis for the numerical solution of partial differential equations for any physical problem of interest.

The elliptic generation system is discussed in Part A, and the operation and use of the codes are covered in Part B.

PART A
ELLIPTIC GENERATION SYSTEM

ELLIPTIC GENERATION SYSTEM

The generation of boundary-fitted coordinates from elliptic systems and the use thereof in the numerical solution of the Navier-Stokes equations is surveyed in Ref. [1]. The foundations of elliptic generation systems are discussed in detail in Ref. [7], and basic configurations of the transformed plane are covered in Ref. [3]. The discussion in this section is an introduction to the subject given by Johnson in Ref. [5] and is incorporated here for convenience.

Basic Ideas

Suppose one is interested in solving a differential system involving two concentric circles, such as shown in Fig. 1, where $r = \text{constant} = \eta_1$ on the inner circle and $r = \text{constant} = \eta_2$ on the outer circle, and θ varies monotonically over the same range over both the inner and outer boundaries, i.e., 0° to 360° .

A cylindrical coordinate system is the obvious choice since a coordinate line, i.e., a line of constant radius, coincides with each boundary. If one now pulls the interior regions between the two circles

apart at $\theta = 0^\circ$ (or $\theta = 360^\circ$) and folds outward, it is easy to visualize the region D_1 becoming the rectangular region D_2 . Likewise, it should be obvious that the right and left sides of the rectangle are reentrant boundaries since $\theta = 0^\circ$ and $\theta = 360^\circ$ are coincident in region D_1 . If one computes a derivative in the cylindrical system at $\theta = 0^\circ$, values at the points marked x and o on both sides might be used. Thus, these same points, as shown in the rectangular region, would be used for a similar derivative in region D_2 . This is the reason for calling these boundaries reentrant boundaries. As shown, the boundary of the inner circle becomes the bottom of the rectangular region while the boundary of the outer circle becomes the top.

The general boundary-fitted system is completely analogous to the system discussed above. In Fig. 2 the curvilinear coordinate, η , is defined to be constant on the inner boundary in the same way that the curvilinear coordinate, r , is defined to be constant on the inner circle in the cylindrical coordinate system. Similarly, η is defined to be constant at a different value on the outer boundary. The other curvilinear coordinate, ξ , is defined to vary monotonically over the same range on both the inner and outer boundaries, as the curvilinear coordinate, θ , varies from 0 to 2π around both the inner and outer circles in cylindrical coordinates. It would be just as meaningless to have a different range for ξ on the inner and outer boundaries as it would be to have θ increase by something other than 2π around one of the circles in cylindrical coordinates. It is this fact that ξ has the same range on both boundaries that causes the transformed field to be rectangular. Note that the actual values of the coordinates, η and ξ , are irrelevant,

in the same way that r and θ may be expressed in different units in cylindrical coordinates.

Now that the values of the coordinates, η and ξ , have been completely specified on all the boundaries of a closed field, it remains to define the values in the interior of the field in terms of these boundary values. Such a task immediately calls to mind elliptic partial differential equations, since the solution of such an equation is completely defined in the interior of a region by its values on the boundary of the region. Thus if the coordinates, ξ and η , are taken as the solutions of any two elliptic partial differential equations, say $L(\xi) = 0$, $D(\eta) = 0$, where L and D represent elliptic operators, then ξ and η will be determined at each point in the interior of the field by the specified values on the boundary. One condition must be put on the elliptic system chosen, since the same pair of values (ξ, η) must not occur at more than one point in the field or the coordinate system will be ambiguous. This condition can be met by choosing elliptic partial differential equations exhibiting extremum principles that preclude the occurrence of extrema in the interior of the field.

This may be illustrated with resort to the governing equation for a stretched membrane. Consider a membrane attached to a flat plate around a closed circuit of arbitrary shape as shown in Fig. 3. Now let a cylinder of arbitrary flat cross section be pushed up through the plate, stretching the membrane upward. The vertical displacement, h , of the membrane will be described by Laplace's equation, $\nabla^2 h = 0$, with $h = h_1$ and h_2 , respectively, on the circuits of contact with the plate and cylinder. If equally spaced grid lines encircling the cylinder had been

drawn on the membrane before displacement, these lines would appear to move closer to the cylinder when viewed from above after displacement of the membrane. None of these lines would cross, however.

Now let pressure be applied on the upper side of the membrane as diagrammed in Fig. 4a. This will cause the slope at the cylinder to steepen, with the effect that the lines will appear to be drawn even closer to the cylinder but still without crossing. This situation corresponds to the Poisson equation, $\nabla^2 h = p$, where p is the applied pressure. If a variable pressure is applied on both sides of the membrane to a sufficient degree, it is possible to make the membrane assume an S shape as shown in Fig. 4b. In this case the encircling lines have crossed, and consequently, a point on the plate can no longer be identified by specifying the encircling line that it lies below (together with a radial ray). This latter case corresponds to a right-hand side of the Poisson equation that is not of one sign over the entire membrane, in which case the extremum principles of Poisson's equation are lost.

Note, however, that if the differential pressure that is applied across the membrane is not too large, the S shape will not be reached. In this case the lines do not cross, but rather the lines seem to concentrate near a line in the interior of the field. Thus the existence of an extremum principle is a sufficient condition to prevent double-valuedness in the coordinate system but is not a necessary condition. Care must be exercised in its absence, however.

Mathematical Development

From the discussion above, a logical choice of the elliptic generating system is Poisson's equation. Thus, based upon Fig. 2, the basic problem is to solve

$$\xi_{xx} + \xi_{yy} = P \quad (1)$$

$$\eta_{xx} + \eta_{yy} = Q$$

with boundary conditions,

$$\xi = \xi_1(x,y) \text{ on } \Gamma_1$$

$$\eta = \text{constant} = \eta_1 \text{ on } \Gamma_1 \quad (2)$$

$$\xi = \xi_2(x,y) \text{ on } \Gamma_2$$

$$\eta = \text{constant} = \eta_2 \text{ on } \Gamma_2$$

The arbitrary curve joining Γ_1 and Γ_2 in the physical plane specifies a branch cut for the multiple-valued function, $\xi(x,y)$. Thus the values of the coordinate functions $x(\xi,\eta)$ and $y(\xi,\eta)$ coincide along Γ_3 and Γ_4 , and these functions and their derivatives are continuous from Γ_3 to Γ_4 . Therefore boundary conditions are neither required nor allowed on Γ_3 and Γ_4 . As previously noted, boundaries with these properties are designated reentrant boundaries.

The functions P and Q may be chosen to cause the coordinate lines to concentrate as desired, in analogy with the membrane discussed above.

As discussed in Ref. [7], negative values of Q result in a superharmonic solution and cause η -lines to move toward the η -line having the lowest value of η , while positive values have the opposite effect. Considering the ξ solution to be superharmonic results in the interior of the $\xi =$ constant lines being rotated in a counterclockwise direction in the physical plane; whereas if the ξ -equation is subharmonic, i.e., P is positive, the lines are rotated in the clockwise direction. These effects are discussed in more detail below. It has been found convenient, as discussed in Ref. [7], to redefine the control functions as

$$P = \frac{1}{J^2} (x_\eta^2 + y_\eta^2) P$$

$$Q = \frac{1}{J^2} (x_\xi^2 + y_\xi^2) Q$$

A major purpose of this coordinate system control is to concentrate lines in viscous boundary layers near solid surfaces, and some automated procedures for this purpose have been developed (cf. Ref. [7]). Control is also useful to improve grid spacing and configuration when complicated geometries are involved.

Since all numerical computations are to be performed in the rectangular transformed plane, it is necessary to interchange the dependent and independent variables in Eq. (1). Using the relations given in Ref. [3], Eq. (1) becomes

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + \alpha P x_\xi + \gamma Q x_\eta &= 0 \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + \alpha P y_\xi + \gamma Q y_\eta &= 0 \end{aligned} \tag{3}$$

where

$$\alpha = x_{\eta}^2 + y_{\eta}^2$$

$$\beta = x_{\xi} x_{\eta} + y_{\xi} y_{\eta}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2$$

$$J = \text{Jacobian of the transformation} = x_{\xi} y_{\eta} - x_{\eta} y_{\xi}$$

with the transformed boundary conditions

$$x = f_1(\xi, \eta_1) \text{ on } \Gamma_1^*$$

$$y = g_1(\xi, \eta_1) \text{ on } \Gamma_1^*$$

$$x = f_2(\xi, \eta_2) \text{ on } \Gamma_2^*$$

$$y = g_2(\xi, \eta_2) \text{ on } \Gamma_2^*$$

Again considering Fig. 2, the boundary functions f_1 , f_2 , g_1 , and g_2 are specified by the known shape of the contours Γ_1 and Γ_2 and the specified distribution of ξ thereon. Boundary data are neither required nor allowed along the reentrant boundaries, Γ_3 and Γ_4 . Although the new system of equations is more complex than the original system, the boundary conditions are specified on straight boundaries and the coordinate spacing in the transformed plane is uniform. Computationally, these advantages far outweigh any disadvantages resulting from the extra complexity of the equations to be solved.

The boundary-fitted coordinate system so generated has a constant η -line coincident with each boundary in the physical plane. The ξ -lines may be spaced in any manner desired around the boundaries by specification of x, y at the equispaced ξ -points on the Γ_1^* and Γ_2^* lines of the transformed plane. As noted above, the entire side boundaries are reentrant boundaries, and thus neither require nor allow specification of x, y thereon.

Now the rectangular transformed grid is set up to be the size desired for a particular problem. Since the values of ξ and η are meaningless in the transformed plane, the η -lines are assumed to run from 1 to the number of η -lines desired in the physical plane. Likewise, the ξ -lines are numbered 1 to the number specified on the boundaries of the physical plane. The grid spacing in both the ξ and η directions of the transformed plane is taken as unity. Second-order central difference expressions are used to approximate all derivatives.

Only one of a pair of reentrant boundaries is considered as a computation line since the (x, y) are equal on both. As an example of how a reentrant boundary is handled, consider the grid in Fig. 5 where "o" indicates a computation point and " Δ " a boundary point. The derivative of x with respect to ξ along $i = 1$ would be written as

$$\left. \frac{\partial x}{\partial \xi} \right|_{1,j} = (x_{2,j} - x_{\text{IMAX}-1,j})/2$$

Again, it should be stressed that all computations are performed on the rectangular field with square mesh in the transformed plane. The resulting set of nonlinear difference equations, two for each point, are solved by accelerated Gauss-Seidel (SOR) iteration using overrelaxation.

Some discussion of this technique is presented in Ref. [8].

It might be noted that both orthogonal and conformal transformations are special cases of the generation of boundary-fitted coordinate systems as the solutions of elliptic partial differential systems. In both of these cases the curvilinear coordinates satisfy Laplace's equation with one coordinate constant on each boundary, and the normal derivative of the other coordinate equal to zero on each boundary. A conformal system also requires a certain relation between the range of the two curvilinear coordinates.

The same procedure may be extended to regions that are more than doubly connected, i.e., have more than two closed boundaries, or equivalently, more than one body within a single outer body. A river reach containing more than one island would be an example. One such transformation for such a problem is illustrated in Fig. 6.

Types of Boundary-Fitted Coordinate Systems

The above discussion of the generation of boundary-fitted coordinates has centered around the idea of using branch cuts to reduce multiply-connected regions to simply-connected ones in the transformed plane. An example using branch cuts is sketched in Fig. 7. Here the body in the field transforms to the entire bottom boundary of the transformed plane, while the entire surrounding boundary, 1 - 2 - 3 - 4 - 5 - 6, transforms to the top boundary of the transformed plane. The sides of the transformed plane are reentrant boundaries, corresponding to the cut, 8 - 1 and 7 - 6, in the physical field. Thus, in the difference equations, points lying just to the right of the right boundary are identical with corresponding points just to the right of the left boundary. This is

the same type of circumstance that occurs with the familiar cylindrical coordinate system, where $\theta = 361^\circ$ is the same point as $\theta = 1^\circ$. Similarly, points just outside the left boundary are coincident with points just inside the right boundary.

Many variations of this type of coordinate system can be produced, cf. Ref. [3]. For instance, the transformed plane corresponding to the same physical field shown in Fig. 7 can be rearranged as shown in Fig. 8. Now the reentrant boundary, corresponding to the cut, is located on a portion of the bottom of the transformed plane. The coordinate lines that result from these two types of arrangements of the transformed plane are shown on each of the figures. As with all the boundary-fitted coordinate systems, the grid is square in the transformed plane regardless of the line configuration in the physical plane.

Multiple-body fields can also be transformed to simply connected regions, an example of which is shown in Fig. 9. Again there are many different possible arrangements of the transformed plane, all of which are created by sliding the boundary segments around the rectangular boundary of the transformed plane. A number of examples are given in Ref. [3] and Ref. [8].

The other type of coordinate system transformation available leaves the multiplicity of the region unchanged. In this case, bodies in the interior of the physical field are transformed to rectangular slabs or even slits in the transformed plane. Three different possibilities are shown in Fig. 10 for the physical plane shown in Fig. 7. In the case of slits, the physical coordinates and solution variables in general have different values at points on the two sides of the slit, even though such

points are coincident in the transformed plane. This does not introduce any approximations, but simply adds a little more bookkeeping to the code. Fields with more than one body in the interior simply result in a like number of slabs and/or slits in the transformed plane.

Comparison of all of the above figures shows that different types of transformation may be more appropriate for different physical configurations. A further example of this is the configuration in Fig. 11, shown with three variations. Generally, the slit/slab form is more appropriate for channel-like physical configurations having bodies in the interior, while the other form works particularly well for "unbounded" regions involving external flow about bodies and for regions having an outer boundary that forms a continuous circuit without pronounced corners around the field. The slab is generally superior to the slit unless the boundary has a sharp point. The case of a single channel without any interior bodies is the same in either form. An example of a river reach containing two islands, using horizontal slits rather than the branch cuts previously presented in Fig. 6, is given in Fig. 12.

Data Required for Generation of Boundary-Fitted Coordinates

The basic input or data required to generate a boundary-fitted coordinate system are the physical coordinates of points on the boundaries. For example, with reference to Fig. 7, the coordinates of points on the body from 8 around to 7 would be required, with these points being spaced in any manner desired as long as there is a continuous progression from 8 to 7. Similarly, the (x,y) values for points on the outer boundary from 1 to 2, etc., on around to 6 would be required. Again these points

may be spaced around the boundary as desired, with no restriction as to how many points lie on each boundary segment, e.g., between 1 and 2 or between 4 and 5, provided that only the total number of points from 1 around to 6 is the same as from 8 to 7. The coordinates of points must be specified on the entirety of these lines. The coordinates of points on reentrant segments of the boundary in the transformed plane, e.g., 1 to 8 and 6 to 7, are not specified but are free to be determined by the solution.

Similarly, with reference to Fig. 10a, the coordinates of outer boundary points are required in the slab/slit transformations. In addition, body points from 6 to 1 on the lower half of the body and from 1 to 6 on the top half are required. No calculations would be made on the slab sides of Figure 10c or slits of Figures 10a and 10b since values at such points are fixed. Points in the interior of a slab are irrelevant. As always, points may be spaced as desired around the bodies and outer boundary segments.

Computer Time Required for Generation of Boundary-Fitted Coordinates

Ref. [8] indicates that the typical time required to generate a one-body coordinate system without coordinate system control (the functions P and Q are set to zero) is about 2 min on a UNIVAC 1106 computer for a 70 x 30 field (70 points on the body). If P and Q are not zero, so that the spacing of coordinate lines is controlled, the computation time increases. Multiple-body coordinate systems typically require about 6 min for a 70 x 40 field. If these same computations were to be made on a CDC-7600 computer, the times quoted above would be reduced by perhaps

an order of magnitude or more. Therefore, the cost of generating boundary-fitted coordinate systems for use in numerical models will be generally insignificant.

COORDINATE SYSTEM CONTROL

Control of the coordinate line spacing in the field can be exercised through the non-zero values given to the Laplacian of the curvilinear coordinates as in Eq. (1), as noted above. With a zero Laplacian, the lines tend to be closely spaced near convex segments and more widely spaced near concave segments. A negative value of the Laplacian causes the lines to move toward lower values of the curvilinear coordinate.

Attraction to Other Coordinate Lines and/or Points

This effect is utilized as in Ref. [8] to achieve attraction of coordinate lines to other coordinate lines and/or points by taking the form of the control functions to be

$$\begin{aligned}
 P(\xi, \eta) = & - \sum_{i=1}^n a_i \operatorname{sign}(\xi - \xi_i) \exp(-c_i |\xi - \xi_i|) \\
 & - \sum_{i=1}^m b_i \operatorname{sign}(\xi - \xi_i) \exp\{-d_i [(\xi - \xi_i)^2 + (\eta - \eta_i)^2]^{\frac{1}{2}}\}
 \end{aligned}
 \tag{5}$$

and an analogous form for $Q(\xi, \eta)$ with ξ and η interchanged. The effects of such control is illustrated in Refs. [7] and [8]. The efficacy of control to improve the accuracy of a physical solution done on the coordinate system has been noted.

In the P function, the effect of the amplitude, a_i , is to attract ξ -coordinate lines toward the ξ_i -line, while the effect of the amplitude

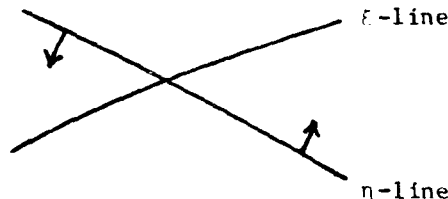
b_1 is to attract ξ -lines toward the single point (ξ_1, η_1) . Note that this attraction to a point is actually attraction of ξ -lines to a point on another ξ -line, and, as such, acts normal to the ξ -line through the point. There is no attraction of η -lines to this point via the P function. In each case the range of the attraction effect is determined by the decay factors, c_1 and d_1 . With the inclusion of the sign changing function, the attraction occurs on both sides of the ξ -line, or the (ξ_1, η_1) point, as the case may be. Without this function, attraction occurs only on the side toward increasing ξ , with repulsion occurring on the other side. A negative amplitude simply reverses all of the above-described effects, i.e., attraction becomes repulsion and vice versa. The effect of the Q function of η -lines follows analogously. It should be noted that P and Q are discontinuous because of the sign function and are equal to sums of second derivatives. As a consequence, the coordinates have continuous first derivatives but discontinuous second derivatives at controlled locations.

In the case of a boundary that is an η -line, positive amplitudes in the Q function will cause η -lines off the boundary to move closer to the boundary, assuming that η increases off the boundary. The effect of the P function will be to alter the angle at which the ξ -lines intersect the boundary, since the points on the boundary are fixed, with the ξ -lines tending to lean in the direction of decreasing ξ . If the boundary is such that η decreases off the boundary, then the amplitudes in the Q function must be negative to achieve attraction to the boundary. In any case, the amplitudes a_1 cause the effects to occur all along the boundary, while the effects of the amplitudes b_1 occur only near selected points on the boundary.

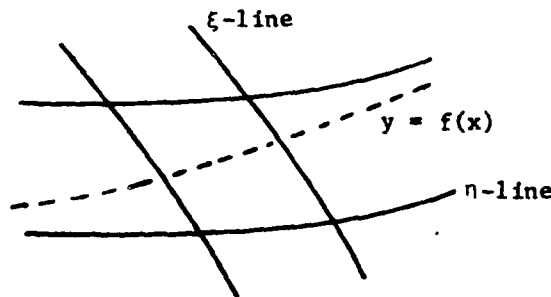
Attraction to Space Curves and/or Points

If the attraction line and/or the attraction points are in the field, rather than on a boundary, then the attraction is not to a fixed line or point in space, since the attraction line or points are themselves solutions of the system of equations, the functions P and Q being functions of the variables ξ and η . It is, of course, also possible to take these control functions as functions of x and y , instead of ξ and η , and achieve attraction to fixed lines and/or points in the physical field. This case becomes somewhat more complicated, since it must be ensured that coordinate lines are not attracted parallel to themselves. The following development was given in Ref. [9].

Recall that in the above discussion, η -lines are attracted to other η -lines, and ξ -lines are attracted to other ξ -lines. It is unreasonable, of course, to attempt to attract η -lines to ξ -lines, since that would have the effect of collapsing the coordinate system:



When, however, the attraction is to be to certain fixed lines in x - y space, defined by curves $y = f(x)$, care must be exercised to avoid attempting to attract η or ξ lines to specified curves that cut the η or ξ lines at large angles. Thus, in the figure below:



it is unreasonable to attract ξ lines to the curve $f(x)$, while it is natural to attract the η -lines to $f(x)$.

However in the general situation, the specified line $f(x)$ will not necessarily be aligned with either a ξ or η line along its entire length. Since it is unreasonable to attract a line tangentially to itself, some provision is necessary to decrease the attraction to zero as the angle between the coordinate line and the given line $f(x)$ goes to 90° . This can be accomplished by multiplying the attraction function by the cosine of the angle between the coordinate line and the line $f(x)$. It is also necessary to change the sign on the attraction function on either side of the line $f(x)$. This can be done by multiplying by the sine of the angle between the line $f(x)$ and the vector to the point on coordinate line.

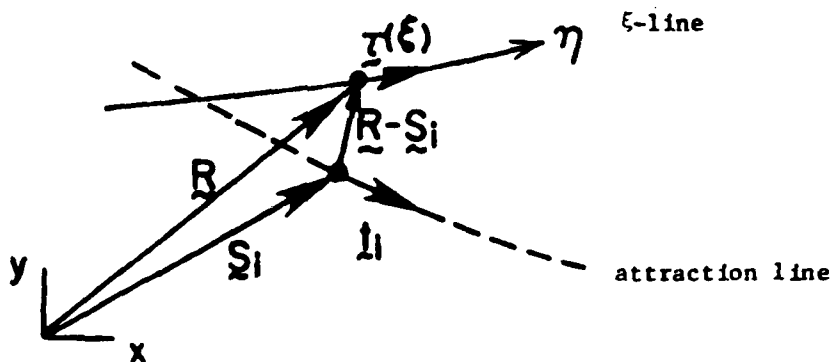
These two purposes can be accomplished as follows. Let a general point on the ξ -line be located by the vector $R(x,y)$, and let the attraction line $y = f(x)$ be specified by the collection of points $S(x_1, y_1)$, $i = 1, 2, \dots, n$. Let the unit tangent to the attraction line be $t(x_1, y_1)$, and the unit tangent to a ξ -line be $\tau^{(\xi)}$.

The control functions $P(x,y)$ and $Q(x,y)$ may then be logically taken as

$$P(x,y) = - \sum_{i=1}^n a_i (t_i \cdot \tau^{(\xi)}) \frac{[\tau_i \times (R - S_i)] \cdot k}{|R - S_i|} \exp(-d_i |R - S_i|) \quad (6)$$

$$Q(x,y) = - \sum_{i=1}^n a_i (t_i \cdot \tau^{(\eta)}) \frac{[\tau_i \times (R - S_i)] \cdot k}{|R - S_i|} \exp(-d_i |R - S_i|)$$

where k is the unit vector normal to the two-dimensional plane. These relations are evident from the figure below:



Here the term $t_i \cdot \tau^{(\xi)}$ serves to decrease the attraction to zero as the angle between the ξ -line and the attraction line approaches 90° . The cross product term changes the sign of the control function on either side of the attraction line to produce attraction on both sides of the line. Again the strength and range of the attraction are determined by the amplitude, a_i , and the decay factor, d_i , respectively.

These functions depend on x and y through both R and $\tau^{(\xi)}$ or $\tau^{(\eta)}$, and thus must be recalculated at each point as the iterative solution proceeds. This form of coordinate control will therefore be more expensive than that based on attraction to other coordinate lines.

There is no real distinction between "line" and "point" attraction with this type of attraction. "Line" attraction here is simply attraction to a group of points that form a line $f(x)$. If line attraction is specified, then the tangent to the line $f(x)$ is computed from the adjacent points on the line. If point attraction is specified, then the "tangent" must be input for each point. The tangents to the coordinate lines are computed from the relations given in Ref. [3].

Control Functions from Boundary-Point Distributions

With the Laplacians of the coordinates equal to zero, the line spacing in the field will not be greatly affected by the distribution of the boundary points, except very near the boundaries. In fact if the control functions are not consistent with the boundary point distribution very large changes in the metric coefficients will occur near the boundaries. Values of the control functions may be determined from the 1D boundary point distribution such that the line spacing in the field will generally follow that on the boundary. This concept was introduced in Ref. [10] and is discussed in Ref. [7] as generalized to 3D in Ref. [11]. However, in the use of control functions that are 1D, it should be noted that excessive concentration of lines can occur near sharp convex corners as discussed in Ref. [7].

With Eq. (3) evaluated in 1D on a straight η -line coincident with the x -axis we have, since $x_\eta = y_\xi = 0$ in this case,

$$\alpha x_{\xi\xi} = -\alpha P(\xi) x_{\xi} \quad (7)$$

The reason for the choice of the form of the control functions in Eq. 3 becomes clear, since α cancels from this equation to leave

$$P(\xi) = -x_{\xi\xi}/x_{\xi} \quad (8)$$

Thus the control function, $P(\xi)$, can be determined from the specified boundary point distribution, $x(\xi)$. Generalizing, x is replaced by arc length along the ξ -line, and the effect will be qualitatively the same when this line is curved. (cf. Ref. [7] for more detail.)

If this value of the control function is then used throughout the field, the ξ -line distribution in the field will generally follow the specified distribution of the end points of these lines on the boundary. With different point distributions on two boundaries, values of the control function $P(\xi, \eta)$ in the field between can be determined by 1D interpolation in η between the values determined in the above manner on the two η -line boundaries. An analogous development applies for the determination of the control function $Q(\xi, \eta)$ from interpolation in between 1D evaluations on two ξ -line boundaries. This interpolation was introduced in Ref. [12] in a 2D coordinate system.

SYSTEM CONFIGURATION

In the present model, the physical field may have both external and internal boundaries of arbitrary shape. The field in the transformed plane is rectangular with rectangular holes corresponding to any internal boundaries. This configuration is illustrated in Fig. 13. Boundary

intrusions may be transformed either to portions of the rectangular outer boundary of the transformed region, as in Fig. 13, or to slabs protruding inward from this boundary as in Fig. 14. A general discussion of possible configurations is given in Ref. [3]. Various outlet shapes and locations, as well as internal obstacles and boundary protrusions such as weirs, can be treated by the same code with only changes in the input. This input consists of the physical cartesian coordinates of the points selected on each segment of the physical boundaries. A small front-end code was written to provide certain line segments (linear, quadratic, and cubic polynomials) with linear or exponential distributions thereon automatically.

The code automatically calculates control functions, $P(\xi, \eta)$ and $Q(\xi, \eta)$, for the coordinate generation equations (3) from the boundary point distribution as discussed above. These functions are calculated from the 1D relations on each boundary segment and are interpolated linearly into the field between opposing boundary sections in the transformed plane.

In addition, attraction of coordinate lines to other coordinate lines and/or points, and to specified lines and/or points in space, also discussed above, is provided through input quantities. This input consists of the coordinate lines and/or points, and the specified space curves and/or points, to which the attraction is to be made and the amplitudes and decay factors for the corresponding attractions.

Several examples of coordinate systems produced by this code are given in Figs. 15-19. Examples of applications of such systems appear in Ref. [4]-[6]. Two further examples, together with complete input listings for the code, follow the description of the code in Part B.

REFERENCES

1. Thompson, J. F., Zahir U. A. Warsi, and C. Wayne Mastin. "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," Journal of Computational Physics, to appear in mid-1982.
2. Thompson, J. F. (ed.). Boundary-Conforming Coordinate Systems, Elsevier/North Holland (1982).
3. Thompson, J. F. "General Curvilinear Coordinate Systems," in Ref. [2].
4. Thompson, J. F. "Numerical Modeling of 2D Width-Averaged Flows Using Boundary-Fitted Coordinate Systems, with Application to Selective Withdrawal from Reservoirs," MSSU-EIRS-ASE-82- , Mississippi State University (1982).
5. Johnson, B. H. and J. F. Thompson. "A Discussion of Boundary-Fitted Coordinate Systems and Their Applicability to the Numerical Modeling of Hydraulic Problems," Misc. Paper H-78-9, U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS (1978).
6. Johnson, B. H. "Numerical Modeling of Estuarine Hydrodynamics on a Boundary-Fitted Coordinate System," in Ref. [2].
7. Thompson, J. F. "Elliptic Grid Generation," in Ref. [2].
8. Thompson, J. F., F. C. Thames, and C. W. Mastin. "TOMCAT - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, 24, 274 (1977).
9. Thompson, J. and W. Mastin. "Grid Generation Using Differential Systems Techniques," Numerical Grid Generation Techniques, NASA Conf. Publication 2166, 37 (1980).
10. Warsi, Z. U. A. and J. F. Thompson. "Machine Solutions of Partial Differential Equations in the Numerically Generated Coordinate Systems," MSSU-EIRS-ASE-77-1, Mississippi State University (1976).
11. Thomas, P. D. "Construction of Composite Three Dimensional Grids from Subregion Grids Generated by Elliptic Systems," AIAA Computational Fluid Dynamics Conference, Palo Alto, 24 (1981).
12. Middlecoff, J. F. and P. D. Thomas. "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA 79-1462, AIAA 4th Computational Fluid Dynamics Conference, Williamsburg (1979).

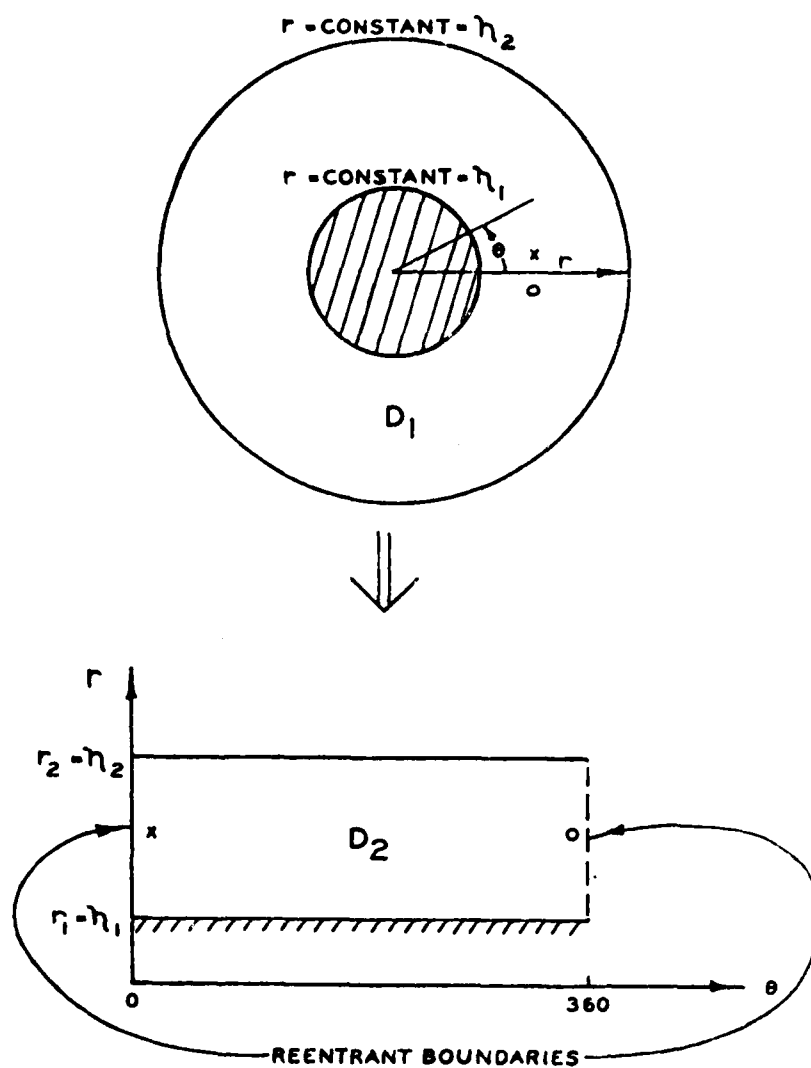


Figure 1. Transformation of domain between concentric cylinders

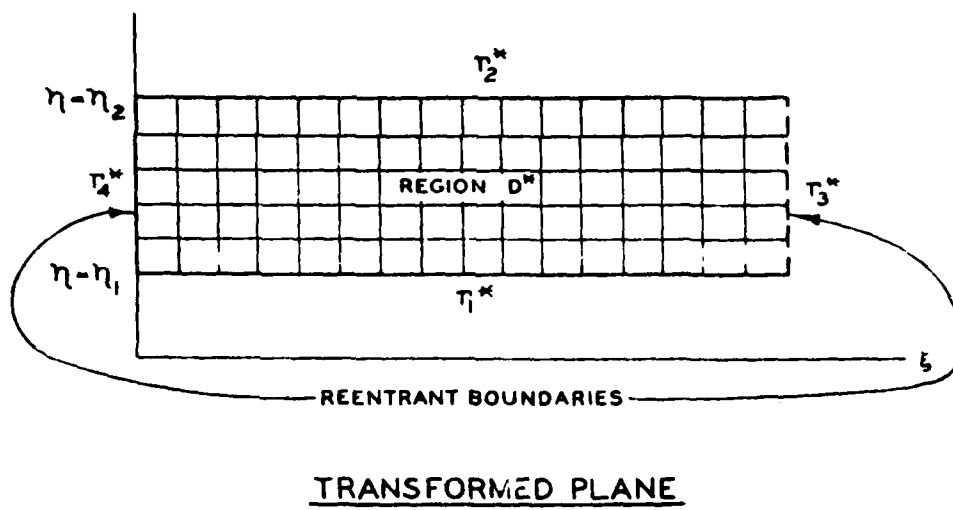
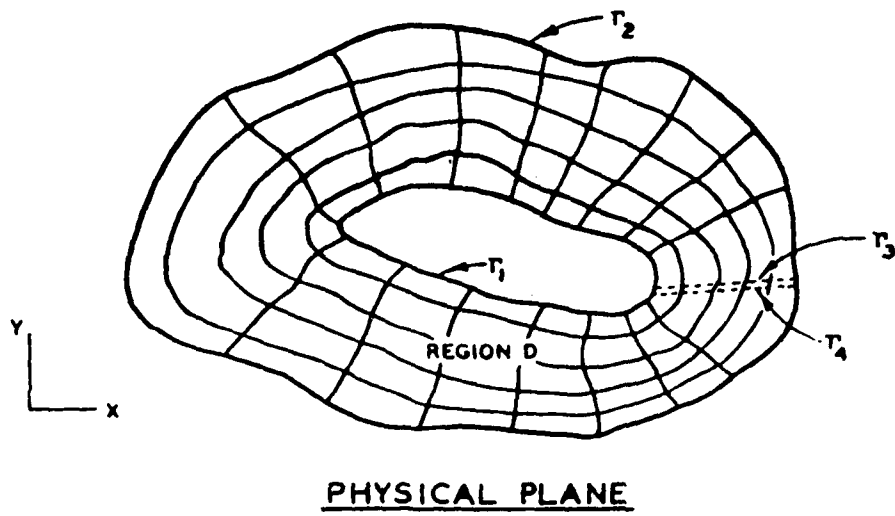


Figure 2. Transformation of an irregular domain

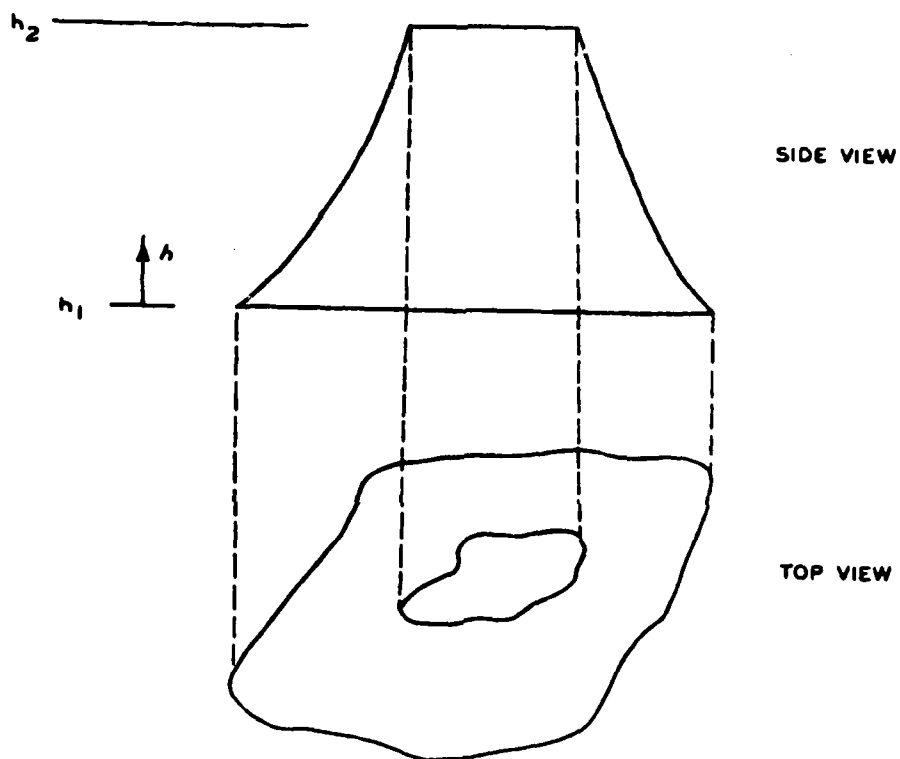
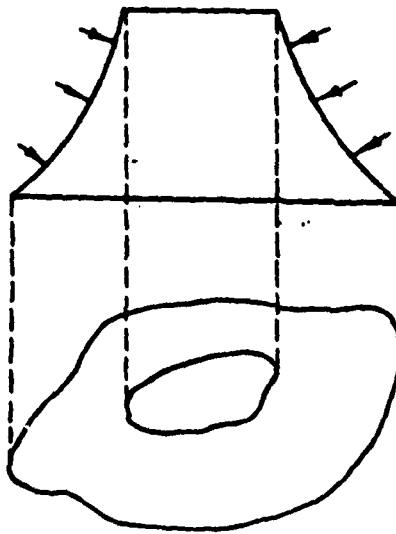
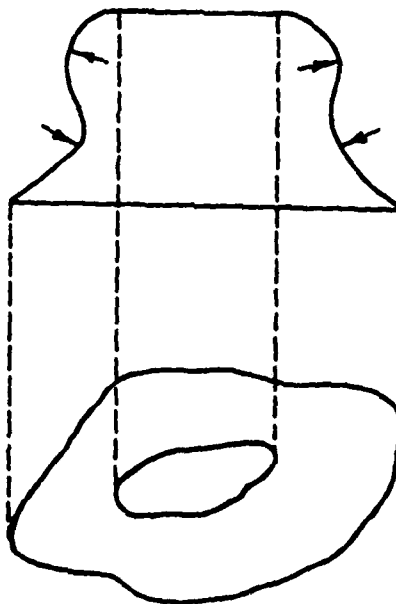


Figure 3. Illustration of extremum principle for Laplace's equation



a.



b.

Figure 4. Illustration of extremum principle for Poisson's equation

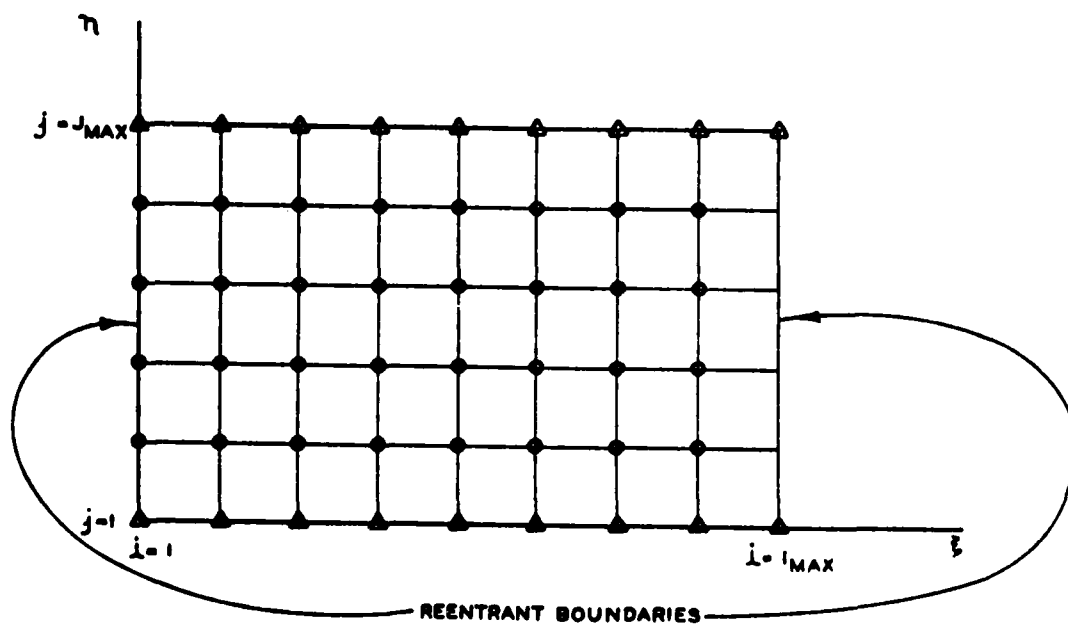
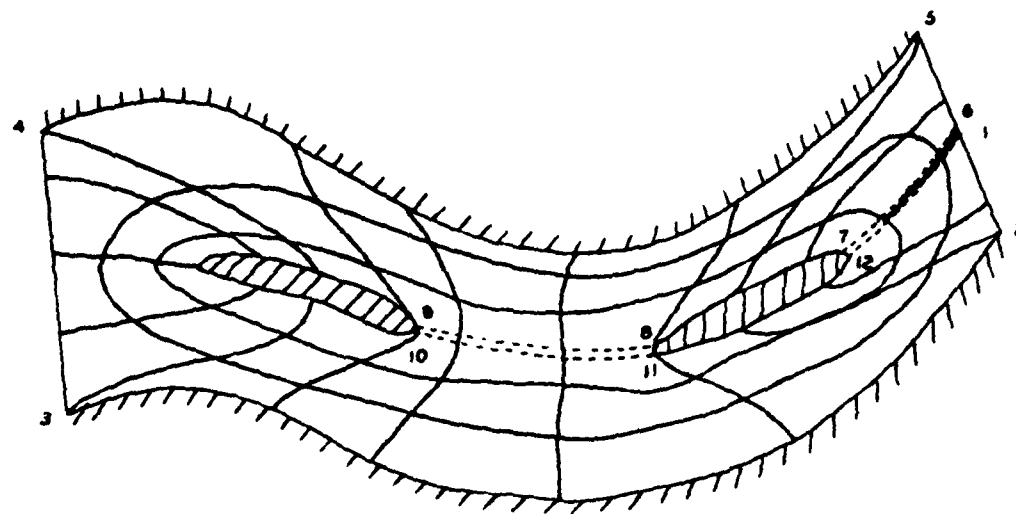
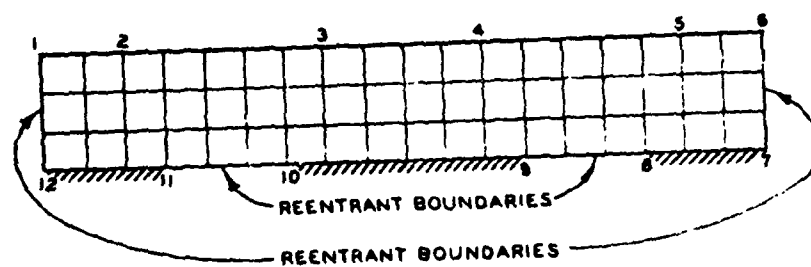


Figure 5. Computational grid in transformed plane

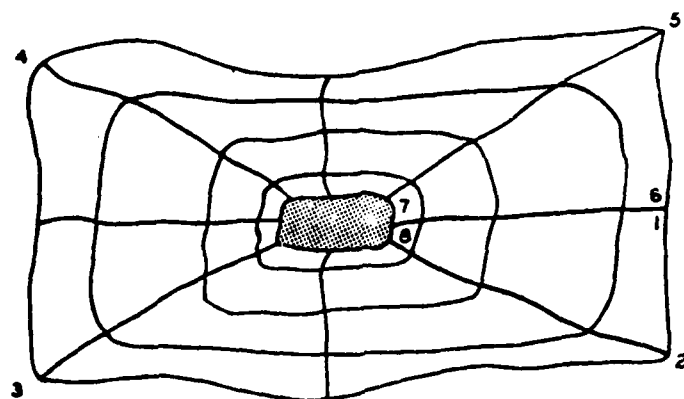


PHYSICAL PLANE

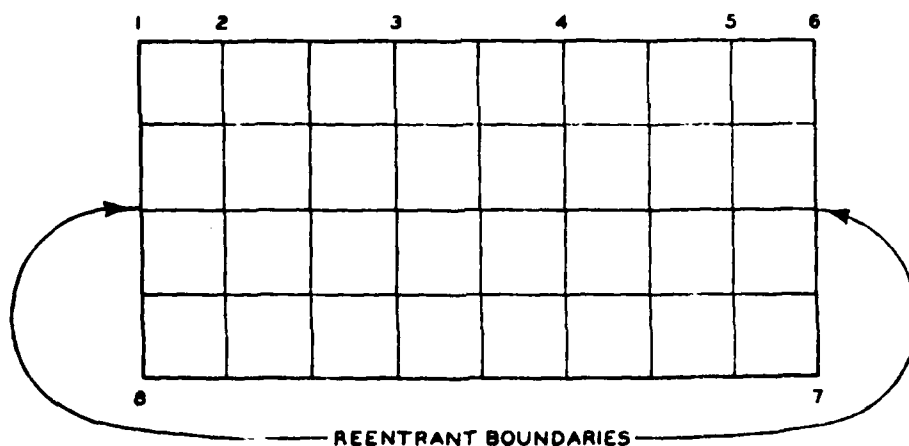


TRANSFORMED PLANE

Figure 6. Boundary-fitted coordinates for a river containing two islands

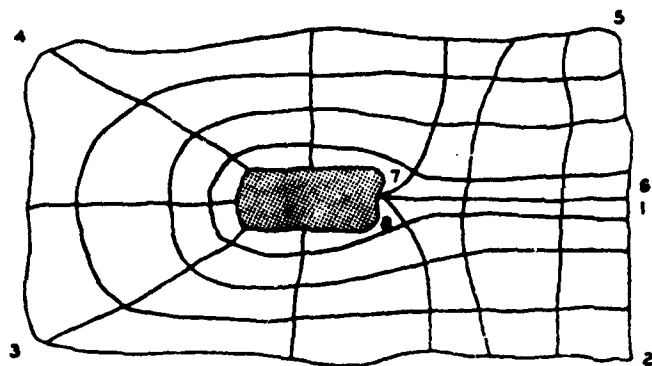


PHYSICAL PLANE

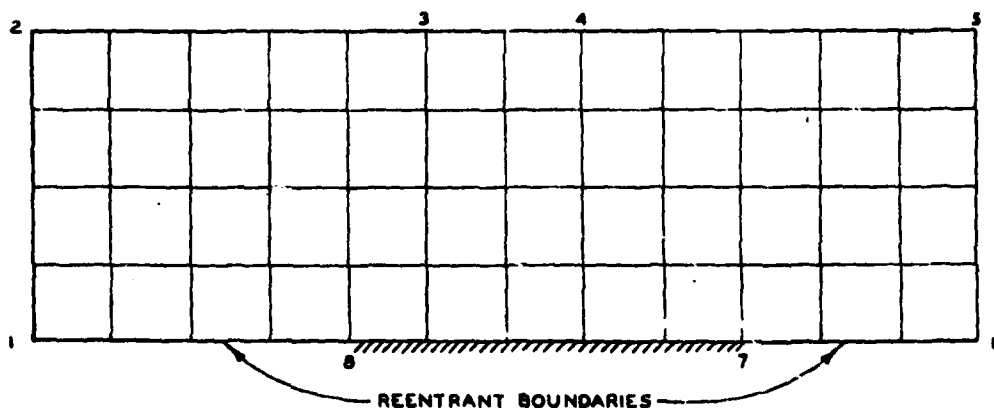


TRANSFORMED PLANE

Figure 7. Example of coordinates generated using a branch cut. Placement of body is such that sides are reentrant boundaries.

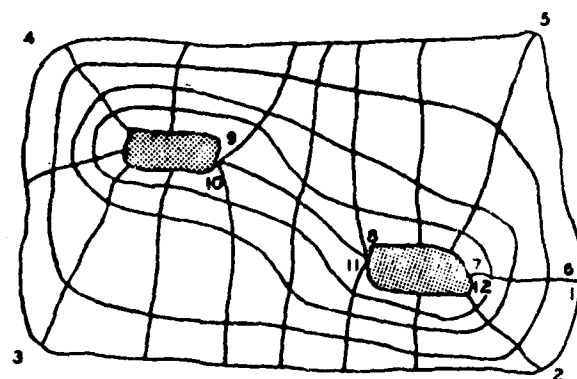


PHYSICAL PLANE

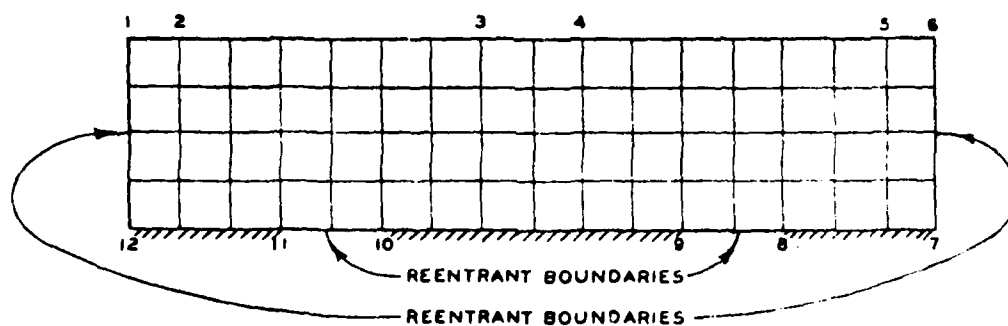


TRANSFORMED PLANE

Figure 8. Example of coordinates generated using a branch cut. Placement of body is such that reentrant boundaries lie on bottom line of the transformed plane.

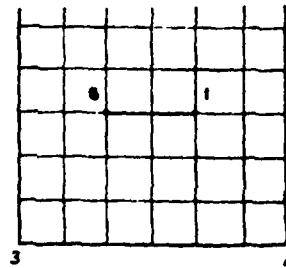
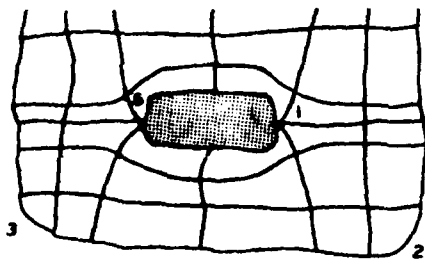


PHYSICAL PLANE

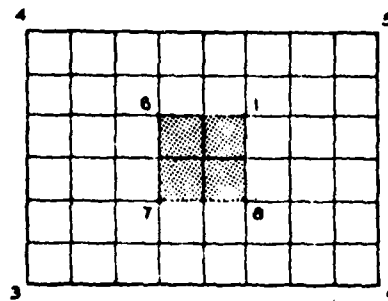
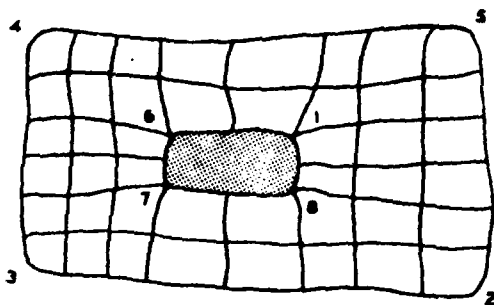


TRANSFORMED PLANE

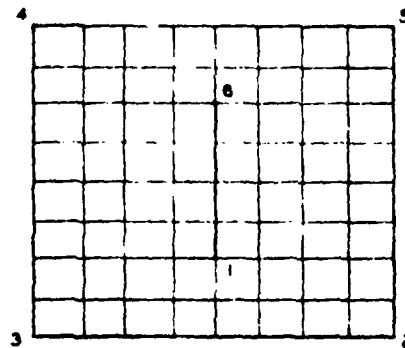
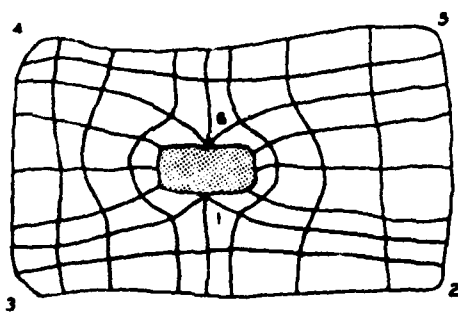
Figure 9. Coordinates generated for a multiple-body field



a.



b.

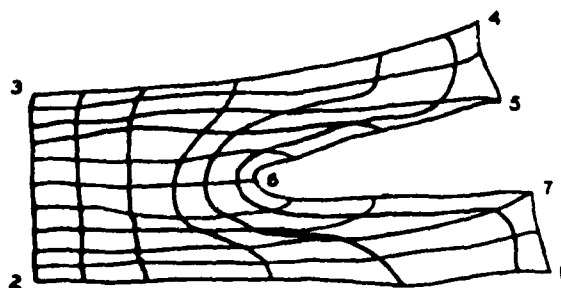


c.

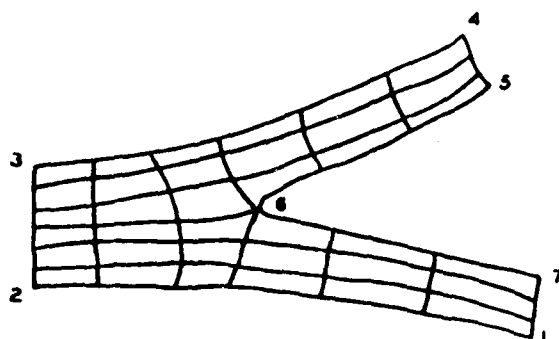
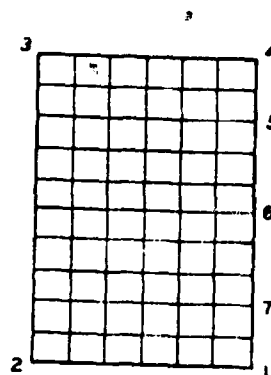
PHYSICAL PLANE

TRANSFORMED PLANE

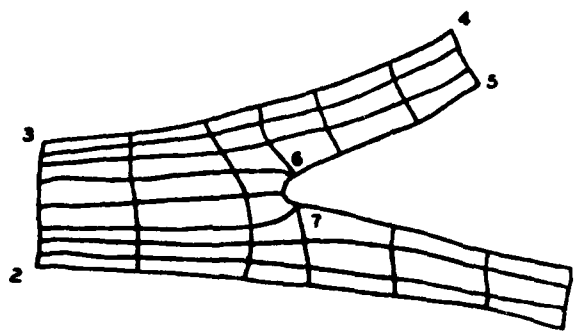
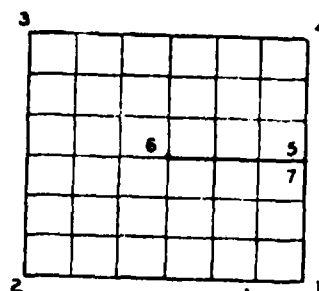
Figure 10. Examples of coordinates generated using slabs/slits



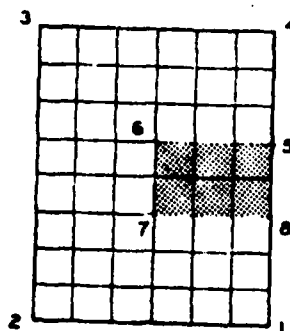
a.



b.



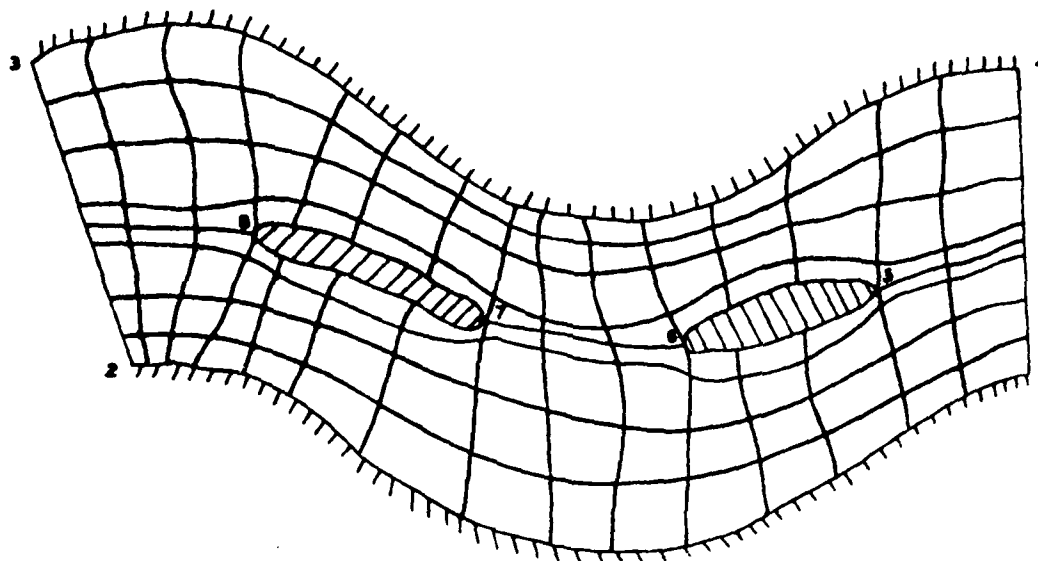
c.



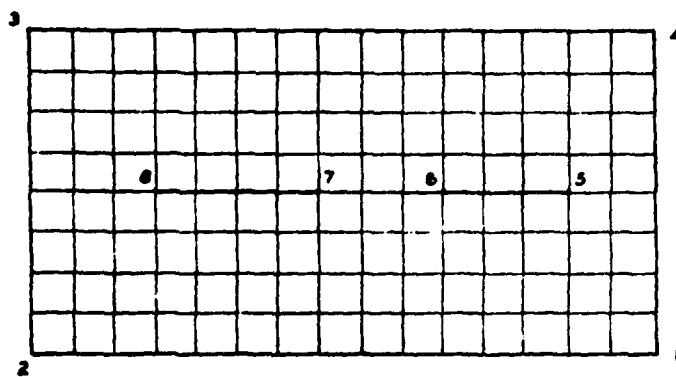
PHYSICAL PLANE

TRANSFORMED PLANE

Figure 11. Comparison of TOMCAT and slit/slab generation of coordinates

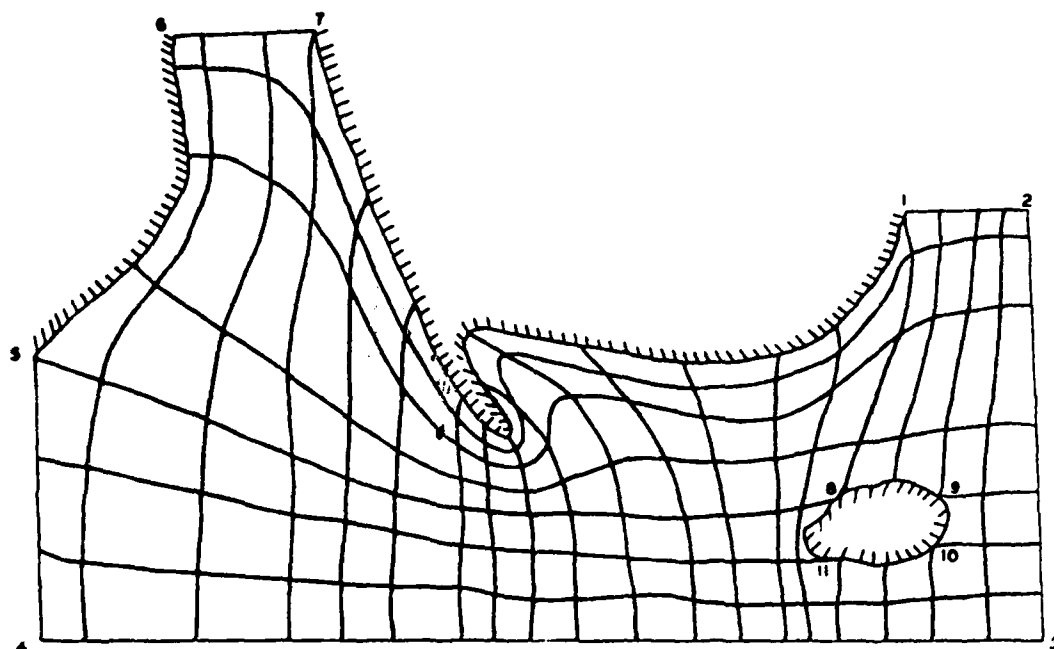


PHYSICAL PLANE

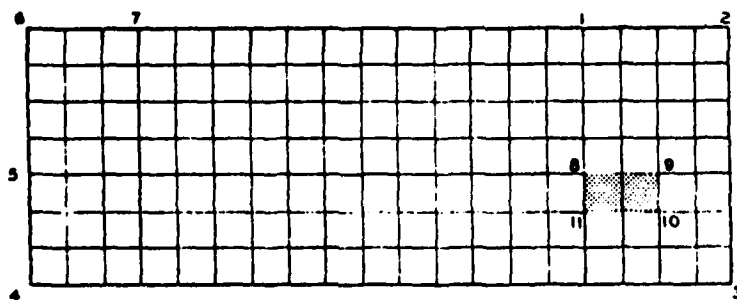


TRANSFORMED PLANE

Figure 12. Coordinates generated with slits for a river with two islands

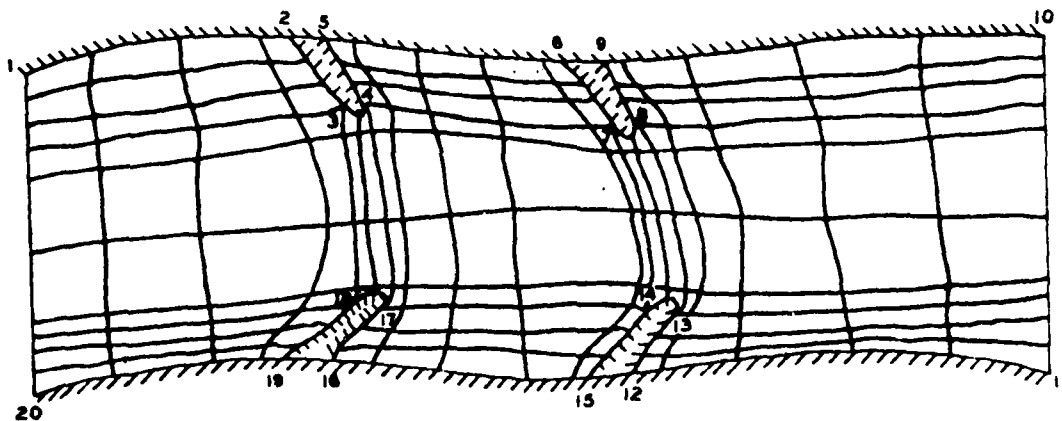


PHYSICAL PLANE

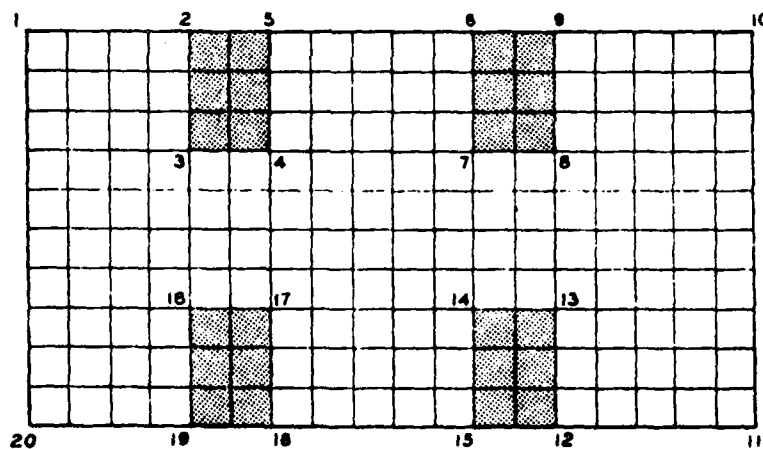


TRANSFORMED PLANE

Figure 13. Example of coordinates generated in a field containing a jetty and an island



PHYSICAL PLANE



TRANSFORMED PLANE

Figure 14. Boundary-fitted coordinates for a river containing dikes

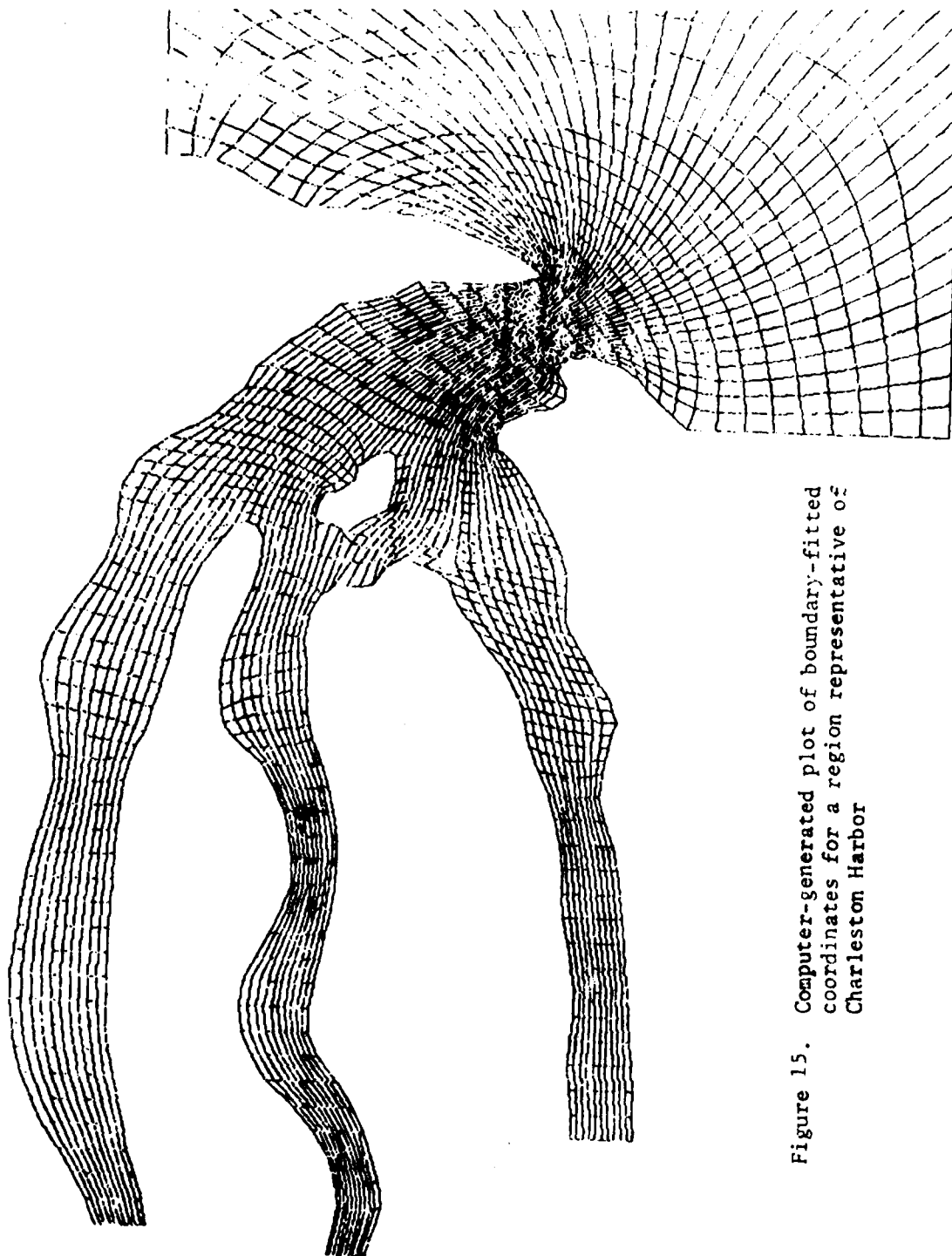


Figure 15. Computer-generated plot of boundary-fitted coordinates for a region representative of Charleston Harbor



Figure 16. Coordinate system for a portion of Lake Ponchatrain.

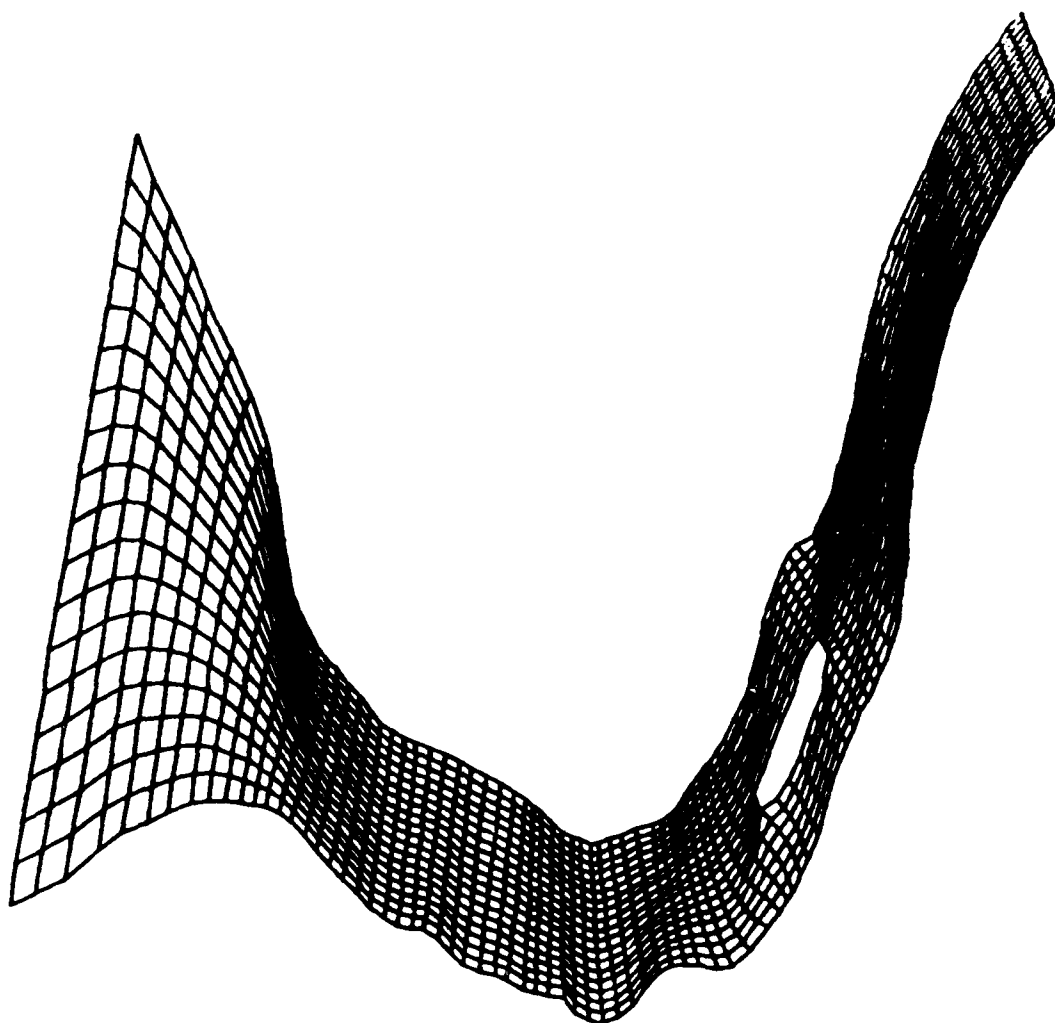


Fig. 17. Hypothetical Estuary Similar to Delaware River
(from B. H. Johnson, Waterways Experiment Station, Vicksburg)

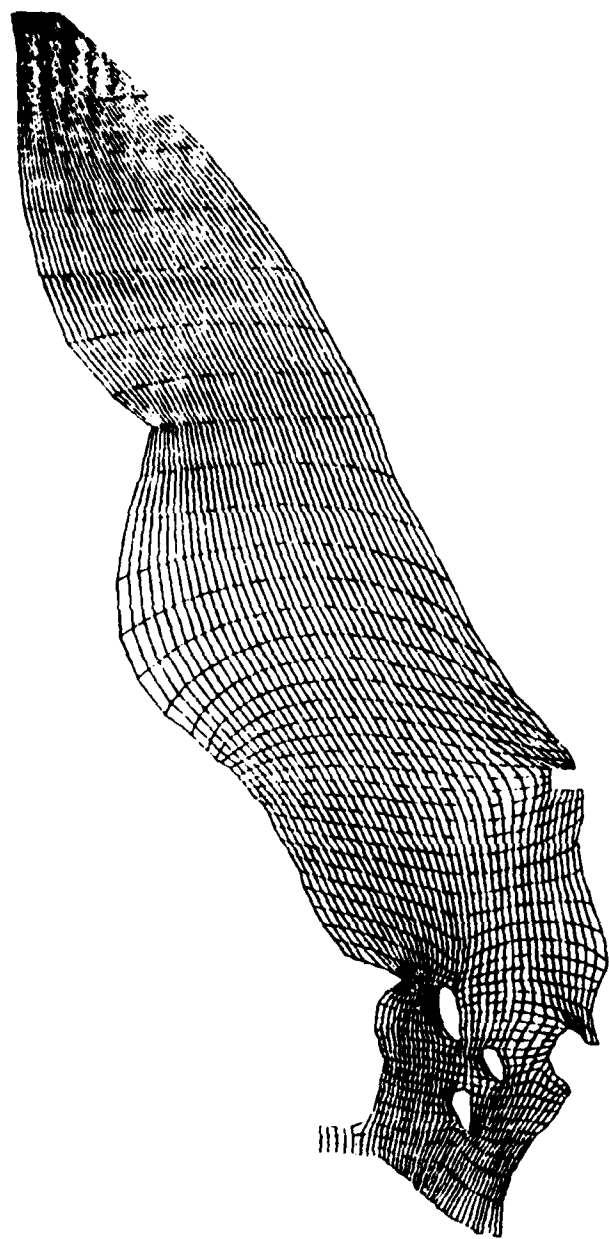


Fig. 18. Coordinate System Fitted to Lake Erie

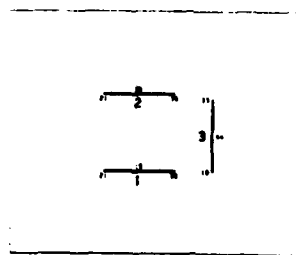
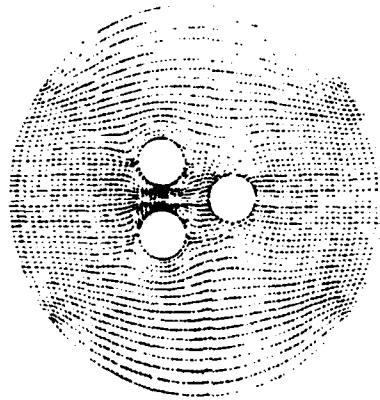


Fig. 19. Transformation to Slits

PART B
COORDINATE CODE

The present code differs from the TOMCAT code described in Ref. [8] in that the latter does not provide for slits and/or slabs in the interior of the transformed plane. Also branch cuts (if used) in the present code are restricted to the entire left and right sides of the outer rectangle in the transformed region. Finally, the present code includes a more extensive means of coordinate line control, involving attraction to space lines/or points and also involving determination from boundary point distributions.

The code for the numerical generation of the boundary-fitted coordinate system from the equations of Part A, together with a front-end code to generate boundary point distributions and a plot code, is discussed below. These codes were implemented on the CRAY-1 computer at the Air Force Weapons Laboratory, Kirtland AFB, New Mexico.

WESCOR (Coordinate System)

This code generates the boundary-fitted coordinate system by solving a set of elliptic partial differential equations by SOR iteration as discussed above in Part A. Attraction of coordinate lines to other coordinate lines and/or points, and to specified lines and/or points in space, is included. The shape and configuration of the boundary are arbitrary, except that the outer boundary must be closed. There may be an arbitrary number of internal closed boundaries transforming to either slits or slabs as discussed in Part A.

The input to this code consists of the point distribution on the boundary of the region, several quantities in connection with the control

of the coordinate line spacing, and the parameters associated with the iterative solution process. This input is described in detail below. The file output from the code LINES can be used directly as a part of the input to this code from file 10.

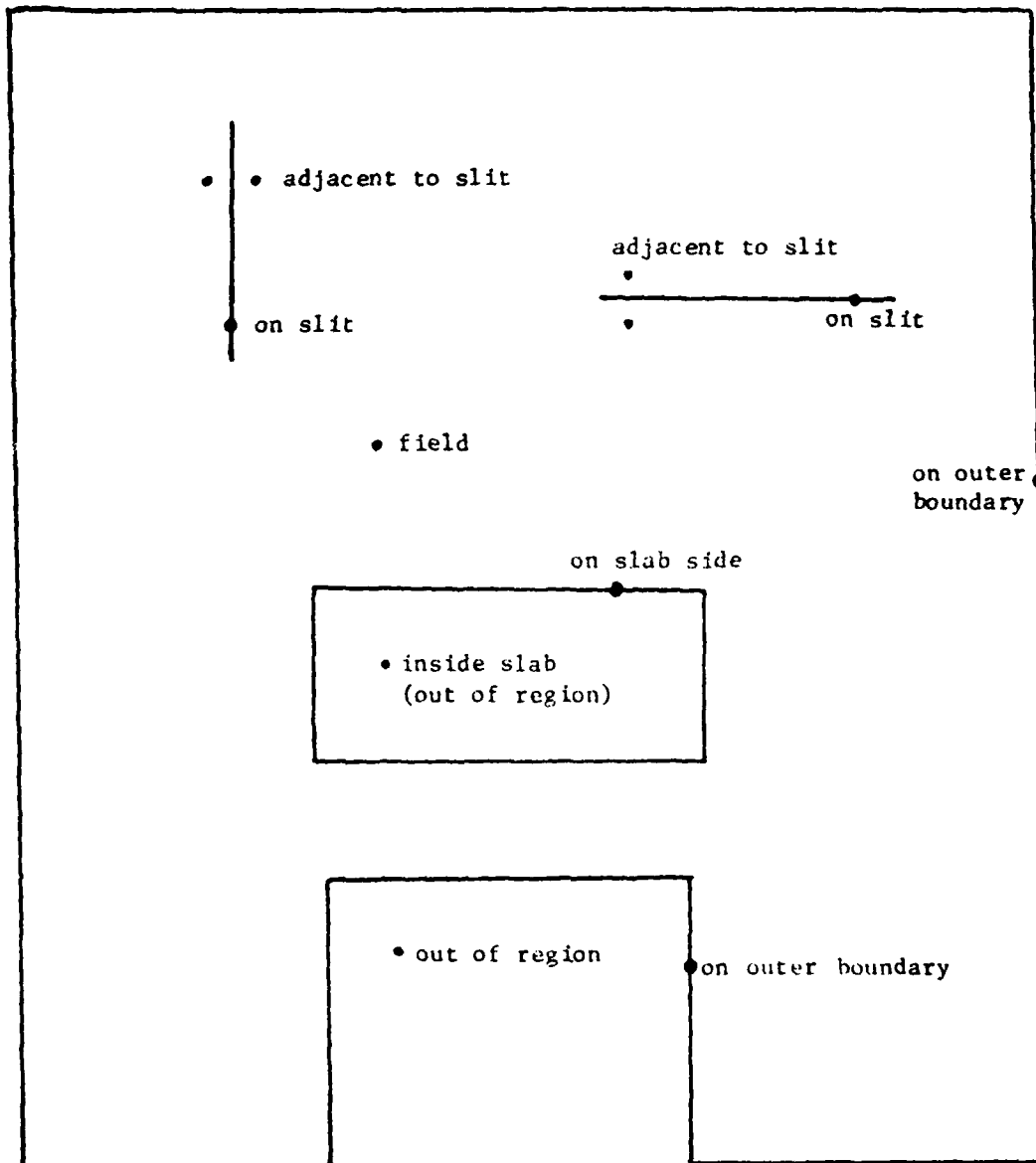
Boundary Configurations

Arrays. The dependent variable field arrays are X and Y , which contain the cartesian coordinates (x,y) for each grid point. The indices (I,J) of these arrays correspond to the curvilinear coordinates (ξ,η) , and run from 1 to IMAX and JMAX, respectively. The increments, $\Delta\xi$ and $\Delta\eta$, in the difference expressions are thus equal to unity by construction. (These increments cancel from all the difference equations and are thus irrelevant.)

In order to treat slit configurations, for which a closed interior boundary in the physical region is collapsed to a slit in the transformed region, there are four other coordinate arrays, XL, YL and XU, YU, which contain the cartesian coordinates on the two sides of the slit. The first index of these arrays corresponds to the location of the point relative to the left end of horizontal slits, or relative to the lower end of vertical slits, this end index being designated unity. The other index identifies the particular slit. For horizontal slits the coordinates on the lower side are in XL and YL, while those on the upper side are in XU and YU. Vertical slits have the coordinates on the left side in XL and YL, and those on the right side in XU and YU.

There is also a field array LSLIT(I,J) containing the point type for each point. This array identifies each point as being on a slit, adjacent to a slit, on a slab side, on an outer boundary, in the field,

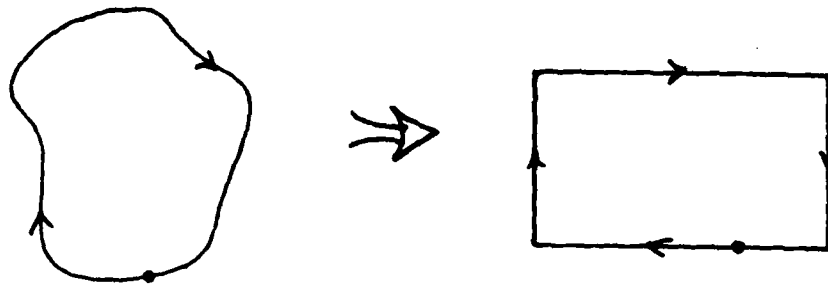
or out of the computation region (inside a slab), as illustrated on the diagram below:



The coordinate system control functions, φ and θ , for each point are contained in the field arrays RXI(I,J) and RETA(I,J), respectively. There are also arrays RXIL, RETAL and RXIU, RETAU, analogous to the array XL, etc., discussed above, which contain the values of these functions on the two sides of the slits. The acceleration parameters for the iteration at each point are in the field array WACC(I,J).

Configuration types. The cartesian coordinates of the points on the entire boundary of the physical region, i.e., the closed outer boundary and any internal boundaries, must be input. There are two basic types of overall configuration included in the code. In one the connectivity of the transformed region is the same as that of the physical region, i.e., the closed outer boundary of the physical region corresponds to a closed outer boundary of the transformed region. With the other type, one branch cut is introduced in the physical region so that the closed outer boundary and one inner boundary of the physical region transform to the bottom and top of a rectangle forming the outer boundary of the transformed region. The left and right sides of the transformed region then correspond to the branch cut in the physical region. Points on these sides therefore are not input but rather are calculated as part of the solution.

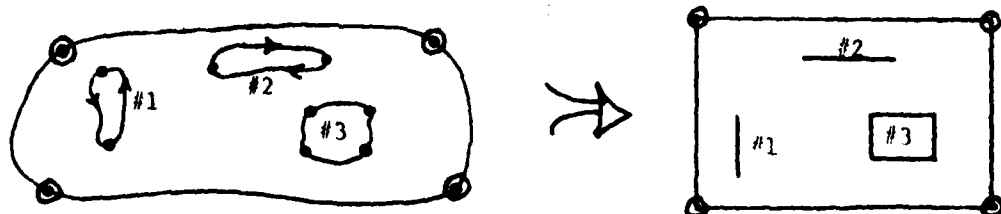
Rectangular outer boundary. If the outer boundary of the physical region is to correspond to a rectangle forming the outer boundary of the transformed region, then the points on this boundary can be input in clockwise succession around the outer rectangle of the transformed region as in the diagram below. If the outer boundary of the physical region



is a circle, then the points on this circle can be generated internally by the code, requiring input only of the radius (YINFIN) and cartesian coordinates of the center ($X0INF, Y0INF$) of the circle, together with the cartesian coordinates of the angular position ($AINFIN$) and indices ($INFXI, INFETA$) of the point at which the clockwise succession of points around the outer rectangle is to start, and the total number of points on the circle ($NINF$). As above, the points will be placed in clockwise succession around the circle or boundary of the physical region and the rectangular boundary of the transformed region. The treatment of the outer boundary is determined by the input parameter IBNDRY.

An alternative procedure for inputting the outer boundary is to input each straight segment of this boundary of the transformed region as a slab side in the manner now to be described for internal boundaries.

Internal boundaries (slits/slabs). Internal boundaries in overall configurations of the former type introduced above correspond to either slits or slabs in the transformed region:



In the case of slits, the points are input in clockwise succession beginning at the right end for horizontal slits or counter-clockwise beginning with the top for vertical slits, and are placed in the arrays XL etc., described above. For slabs, the four sides are input independently and the succession of points may be in either direction on each side. In fact, it is not even necessary for the four sides of one slab to be input in succession; the sides of all slabs in the field may be placed in any order in the input. The coordinates of the points on slab sides are placed directly in the field arrays X and Y. This input of boundary segments corresponding to slits or slabs is accomplished as follows.

For horizontal slits, the ξ -indices(I) of the left and right ends are placed in the arrays LB1 and LB2, respectively. The η -index (J) of the entire slit or slab side is placed in the array LB3. In the case of vertical slits, the η -indices (J) of the bottom and top go in LB1 and LB2, while the ξ -index (I) goes in LB3. Slab sides are treated in the same manner except that, since the points thereon may be input in either direction, LB1 and LB2 contain the indices of the end points of the side in either order, i.e., LB1 may exceed LB2. The points are input from LB1 to LB2.

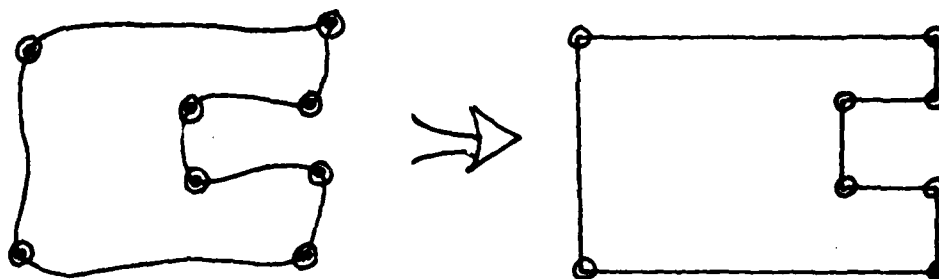
For both slits and slab sides, a flag is placed in an array LTYPE to designate the segment as a slit or slab side in horizontal or vertical orientation:

- +1 horizontal slit
- +2 vertical slit
- 1 horizontal slab side
- 2 vertical slab side

The code computes the number of points on the slit or slab side from the

values of LB1 and LB2 and places this value in the array LPT. All of these arrays are single-dimension arrays, there being one set of parameters for each slit or slab side. The total number of slits and slab sides, including those on the outer boundary as described below, is specified by the input parameter NBDY.

Outer boundary intrusions. As noted above, the outer boundary can be input in segments as slab sides. This is illustrated below.



This is done just as described above for internal boundaries except that values of -11 and -12, respectively, are input for LTYPE for horizontal and vertical segments of the outer boundary.

Branch cut. With the other type of overall configuration, involving a branch cut, the outer boundary and the internal boundary connected to the cut are both input clockwise from the points joined by the cut. As noted above, these points are placed on the top and bottom of the rectangle forming the outer boundary of the transformed region. This type of configuration is elected through the input parameter NREN. Additional internal boundaries can be input as either slits or slabs exactly as described above.

Boundary input. Provision is made for reading the boundary points either from card images, (x and y for one point to a card in 2F10.0

format) or from the output of the LINES code described below, as determined by the input parameter ISLIT. The outer boundary must be input as segments of slab sides if this boundary is included on the output of the LINES code.

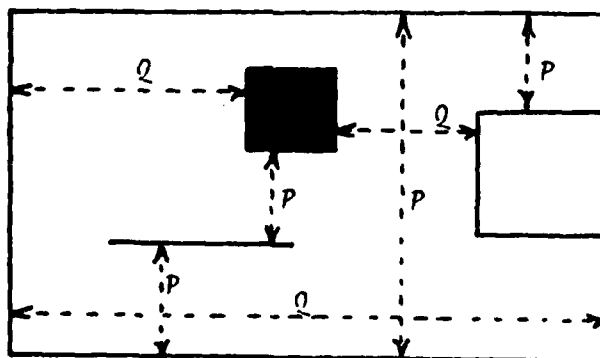
Control Functions

Coordinate system control is included through both the attraction of coordinate lines to other coordinate lines and/or points and to specified lines and/or points in the physical region, as described in Part A. (For completeness, provision is made for repulsion as well as attraction.)

Attraction to coordinate lines and/or points. The first of these requires the input of the index (indices) of the curvilinear coordinate line, together with the associated attraction amplitude and decay factor, for each line (point) to which the attraction is made. For attraction to lines, the index, amplitude, and decay factor are placed in the arrays JLN, ALN, and DLN, respectively, while for attraction to points, the corresponding arrays IPT, JPT, APT, and DPT are used.

Attraction to space lines and/or points. For attraction to specified lines and/or points in space, the input is similar in regard to the amplitude and decay factors, using the arrays APT and DPT. It is necessary, of course, to also input the cartesian coordinates of the points on the line, or the isolated points, to which the attraction is made. These coordinates are placed in the arrays XPT and YPT. For attraction to points, it is also necessary to input the components of a vector normal to the desired direction of the attraction for each point, these components being placed in the arrays VEC1 and VEC2.

Effect of boundary point distribution. In addition to the above types of attraction, the control functions also include the effect of the boundary point distribution discussed in Part A. This is done by evaluating one of the control functions on each boundary segment in the transformed region (P on η - lines, Q on ξ - lines) from the one-dimensional relations in terms of arc length discussed in Part A. These values are placed in the arrays RXI and RETA, except for slits where the arrays RXIL, etc., are used in the manner described above for XL, etc. Values of the control functions in the field are then interpolated linearly between facing boundary segments, P being interpolated vertically and Q horizontally, as illustrated in the following diagram.

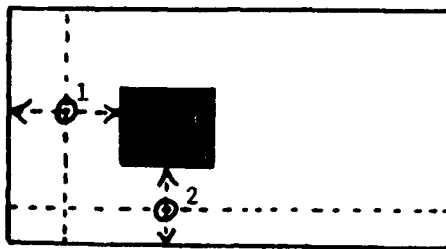


This evaluation is done first and then the contributions to the control functions from the line and point attraction is added to the arrays RXI and RETA in the field.

Iterative Solution

Initial guess. The initial guess for the values of the cartesian coordinates in the field, i.e., the values in the arrays X and Y in the field, that is used to start the iterative solution is obtained by the

same type of interpolation between facing segments described above for the control functions, except that both X and Y are interpolated between the pair of facing segments with the smallest separation in the transformed region. Thus values at point 1 in the figure below would be obtained by horizontal interpolation, but at 2 the interpolation would be vertical:



Since very strong control functions can sometimes make the convergence of the iterative solution difficult in complicated configurations, provision is made for first converging the field with the control functions set to zero and then re-converging in steps as these functions are increased to full value. Actually this feature is rarely needed.

Acceleration parameters. As discussed in Part A the solution for the cartesian coordinates in the field is done by SOR iteration. Either a uniform value of the acceleration parameter can be input as R(1) or the code will calculate a locally optimum value at each point in the field, these values being placed in the field array WACC. This calculation is discussed in Ref. [8], where it is noted that the values obtained are not truly optimum in all cases. Therefore this provision has not been found to be as generally efficient as simply using a uniform value, since the calculation of the acceleration parameter involves a square root and hence is time consuming. The uniform value should be around 1.85 for

large fields. This value should be decreased for strong control functions or small fields.

Iterative process. The iteration continues until either the magnitude of the changes in the values of x and y at each point in the field between iterations is less than the tolerances input as $R(2)$ and $R(3)$, respectively, or until the maximum number of iterations allowed (input as $ITER$) is reached. In the latter case the partially converged solution is stored on file 10 for restart. The input parameter $IDISK$ can cause the code to read this partially converged solution from file 10 and continue the iterations. This parameter also controls the disposition of the final solution, which is normally stored on file 11 for use in the flow solution, but can be simply printed without being stored if desired. Various other input parameters, such as print options, etc., are explained in the detailed input instructions given below and in the source listing.

Code Operation

Initial input and setup. The code uses the values of $NDIM$, $NDIM1$, $NDIM2$, and $NDIM3$, which are assigned by a $DATA$ statement, to determine if the problem specified by the input will fit in the arrays as dimensioned. The first two of these parameters, $NDIM$ and $NDIM1$, correspond to the dimensions of the field arrays, X , etc. The last two, $NDIM2$ and $NDIM3$, correspond to the dimensions of the slit arrays XL , etc. The last parameter, $NDIM3$, also corresponds to the dimension of the segment arrays $LB1$, etc. Thus $NDIM$ is the maximum value of ξ that can be used, while $NDIM1$ is the maximum value η allowed. Also $NDIM2$ is the maximum number of points that can be used on a slit or slab side, and $NDIM3$ is

the maximum number of slits and slab sides that can be used. The input thus must satisfy the following:

$$IMAX \leq NDIM$$

$$JMAX \leq NDIM1$$

$$|LB2(L) - LB1(L)| + 1 \leq NDIM2 \quad L = 1, 2, \dots, NBDY$$

$$NBDY \leq NDIM3$$

After the initial input parameters are read, the code does some setup of various intermediate parameters and checks for compatibility with the array dimensions. The value of IDISK is then checked to determine if the solution is to be started from the beginning or if a partially converged solution is to be continued.

Boundary input and construction. If the start is from the beginning, the point type array LSLIT is initialized to -20000 on the outer rectangle formed by $I = 1$ & $IMAX$ and $J = 1$ & $JMAX$, and to 0 inside this rectangle.

Next the points on the slits and/or slab sides (if any) are read from either card images or file 10. Points on slits are placed in the slit arrays XL, etc., while points on slab sides are placed directly in the field arrays X and Y. The point type array LSLIT is set to $-(10000 + L)$ at points on slab sides, where L identifies the particular segment in the order as input, unless the side is a part of the outer boundary in which case LSLIT is left at -20000. At the same time, 10 is added to the segment type array LTYPE for slab sides on the outer boundary, resulting in replacing the input values of -11 and -12 with -1 and -2, respectively, in conformance with the usage for slits.

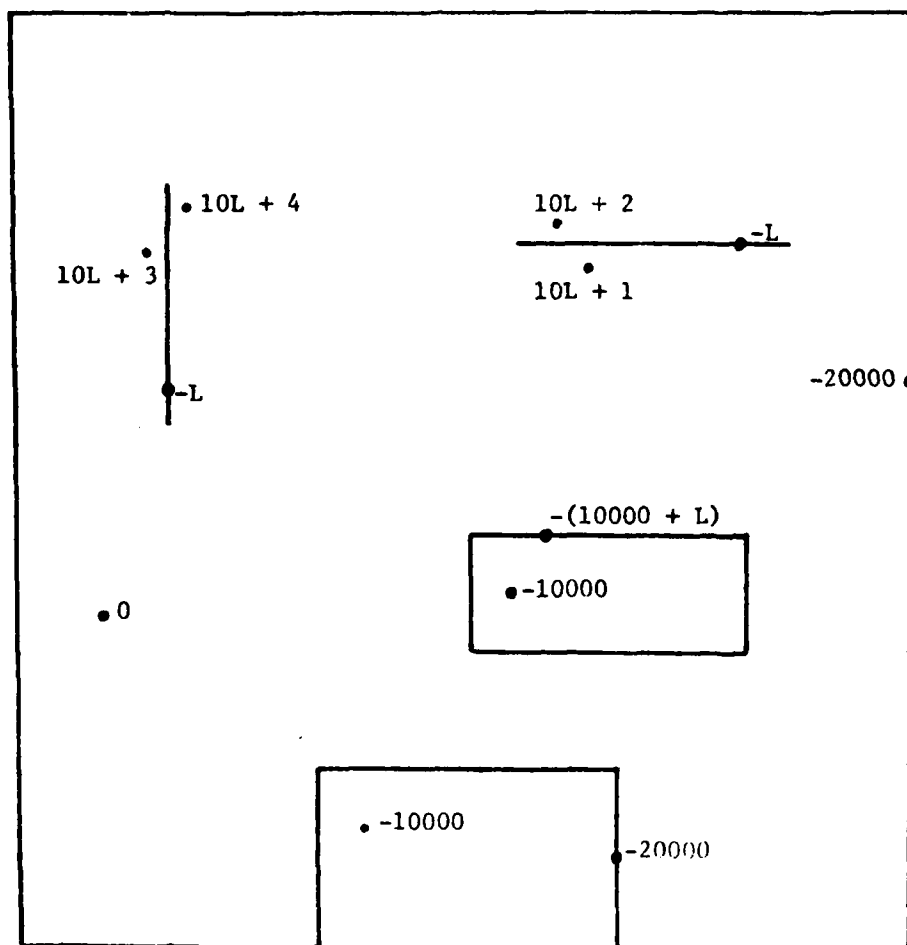
The slit arrays, XL, etc., (if any) are then printed and subroutine BNDRY is called for the outer boundary. If the outer boundary is not

input in segments as slab sides, this boundary is either input as a succession of points proceeding from a specified point completely around the outer rectangle formed by $I = 1, I_{MAX}$ and $J = 1, J_{MAX}$, or a circular outer boundary is generated internally and placed on this rectangle. Both of these procedures are performed by this subroutine by calling the subroutine INFBDY, which either reads a point from a card image or calculates a point on the circle.

Point types. Next the point type array LSLIT is set to the following values on and adjacent to slits (if any). Here L identifies the particular slit in the order as input:

-L	:	on slit	
10L + 1	:	below horizontal slit	not adjacent to slit ends
10L + 2	:	above horizontal slit	
10L + 3	:	left of vertical slit	
10L + 4	:	right of vertical slit	

The point type array LSLIT is then set to -10000 for points outside the computational region, i.e., inside slabs, by sweeping along each ξ and η line and noting when the computational region is entered or left across a slab side. The complete point type array then contains the values indicated in the diagram below:



Control functions and initial guess. With all of the boundary points in place and the point type array filled, the code then calls subroutine `CONTRL` to evaluate the control functions on the entire boundary (including internal boundaries). The subroutine `GUESSA` is called next to calculate the control functions and the initial guess for the cartesian coordinates in the field by interpolation from the values on the boundaries. This interpolation is done at each point in the field by locating the pair of

boundary segments facing the point (one or both members may be internal boundaries) and interpolating between these segments. For the coordinate values, the distances separating the pair of segments facing the point in the horizontal and vertical directions are examined and the interpolation is done between the pair with the smaller separation.

Iterative solution. If the solution is to be restarted from a partially converged result, then all of the above computations are skipped and the partially converged solution is read from file 10 instead. In either case the initial array values are printed at this point according to the input print controls.

Subroutine TRANS is now called to perform the iterative solution. This subroutine first reads the parameters associated with the attraction of curvilinear coordinate lines to other curvilinear coordinate lines and/or points. The species of line being controlled, i.e., ξ or η , is read into ATYP, and whether the control is to be attraction or repulsion is determined by the input parameter ITYP. The number of coordinate lines and points designated as sources of attraction are read into NLN and NPT, respectively. Also a common decay factor and a common amplitude multiplication factor to be used for all attraction lines and points for this species can be read into DEC and AMPFAC, respectively.

For each species of control, subroutine RHS is called to read the attraction line index, or point indices, and the amplitude and decay factor for each. This subroutine also sums the effects for all such attraction lines and points and adds this cumulative effect to the control function at each point in the field in accordance with Eq. (5) of Part A.

Subroutine TRANS then reads the parameters associated with attraction of curvilinear coordinate lines to specified lines and/or points in space and adds the cumulative effect of all such attraction lines and/or points to the control functions at each point in the field. This is done in a similar manner as described above. Subroutine RHSXY reads the cartesian coordinates of the points on the specified attraction line and those of the isolated attraction points and calculates the normal to the attraction line. These quantities are placed in the arrays XPT, YPT, VEC1, and VEC2. The addition to the control functions in this case must be changed as the iterative solution of x and y proceeds since the control functions depend on x and y for this type of attraction.

After completing the calculation of the control functions, subroutine TRANS reads the parameters that provide for a gradual implementation of these equations during the iteration, and performs some setup for the iterative solution.

The field is then swept iteratively until convergence is achieved or the maximum number of iterations allowed is reached. In each iteration, new values for x and y at points having the point type LSLIT non-negative are calculated.

First the coordinate derivatives are calculated, and the Jacobian and other such quantities and coefficients are evaluated. Then the locally optimum acceleration parameters are calculated if such is elected. The change in these acceleration parameters between iterations is monitored and the values are frozen when the magnitude of the change falls below a specified tolerance at all points. (This change between iterations, and the analogous changes in x and y, are calculated by calling subroutine ERROR). The acceleration parameter is placed in the field

array WACC. The addition to the control functions from attraction to specified lines and/or points in the physical region is calculated next, and then the new values of x and y for the point are calculated.

This procedure is followed for all points in the field, i.e., points having the point type LSLIT non-negative. For points adjacent to slits it is necessary to obtain the values on the slit from the slit arrays, XL, etc., and the calculations are done in that case by calling subroutine SLIT.

After each sweep of the field the maximum changes in x and y from the previous sweep are compared with the input tolerances. If the maximum number of iterations allowed by the input is reached before convergence, then the partially converged solution is written on file 10 for potential restart. If convergence is obtained the solution is written on file 11.

LINES (Boundary Segments)

The small front-end code LINES generates a distribution of a specified number of points on a curve between two specified points. The curve may be specified to be a straight line, a circular or elliptic arc, a quadratic with zero slope at either end point, or a cubic with the slope specified at both ends. In any case the point distribution on the curve may be uniform or exponentially concentrated toward either end. The input consists of the number of curves to be generated and, for each curve, the number of points on the curve, the type of curve, the end points, and the particular quantities to be specified in connection with each curve. Detailed instructions for input are given below.

The cartesian coordinates of the points generated on each curve are output in succession on file 10 by a separate unformatted write statement for each point (WRITE(10) X,Y). Since more than one curve can be generated in one run, this code can be used to build an entire boundary composed of segments of different types. The generation of the curves and the exponential concentration of points thereon are explained in the following section.

Generation of Curves

Straight line. Here we have simply

$$y = a + bx$$

so that with the end points, (x_1, y_1) and (x_2, y_2) , specified we have

$$\begin{vmatrix} 1 & x_1 \\ 1 & x_2 \end{vmatrix} \begin{vmatrix} a \\ b \end{vmatrix} = \begin{vmatrix} y_1 \\ y_2 \end{vmatrix}$$

so that

$$a = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$$

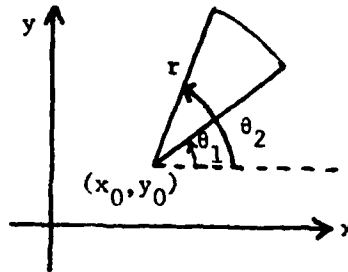
$$b = \frac{y_2 - y_1}{x_2 - x_1}$$

Circular arc. For a circular arc of radius r centered at (x_0, y_0) with θ measured counter-clockwise from the positive x-axis, we have

$$x = x_0 + r \cos \theta$$

$$y = y_0 + r \sin \theta$$

The end points are defined by inputting the radius r and center of the arc (x_0, y_0) , together with the angles, θ_1 , and θ_2 , of the end points.



Elliptic arc. In this case we have, for an ellipse with semi-major axis, a , and semi-minor axis, b , centered at x_0, y_0 , the equation

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

which can be written in terms of the angle θ , measured counter-clockwise from the positive x -axis, and the angular-dependent radius $r(\theta)$ as

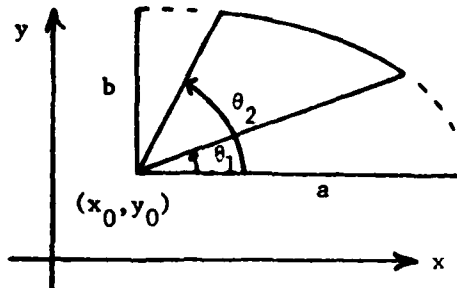
$$x = x_0 + r(\theta) \cos \theta$$

$$y = y_0 + r(\theta) \sin \theta$$

Then

$$r(\theta) = \left| \frac{\cos^2 \theta}{a^2} + \frac{\sin^2 \theta}{b^2} \right|^{-\frac{1}{2}}$$

The end points are specified by inputting the axes, a and b, the center (x_0, y_0) , and the angles of the end points.



Quadratic with zero slope at end point. Here we have

$$y = a + bx + cx^2$$

$$y' = b + 2cx$$

Then with the end points, (x_1, y_1) , and (x_2, y_2) , specified together with the specification of zero slope at end point i ($i = 1$ or 2) we have

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 0 & 1 & 2x_1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ 0 \end{pmatrix}$$

which is solved for the coefficients a, b, c.

Cubic. The cubic equation is

$$y = a + bx + cx^2 + dx^3$$

$$y' = b + 2cx + 3dx^2$$

or

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_2 & 3x_2^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_1' \\ y_2' \end{pmatrix}$$

which is solved for the coefficients a, b, c, d.

Exponential Concentration of Points

The exponential distribution of points on the curve of any type is done by taking

$$x = x_1 + (x_2 - x_1) \left[1 - \frac{1 - e^{-\alpha(N-n)}}{1 - e^{-\alpha(N-1)}} \right]$$

for concentration near the first end point and

$$x = x_1 + (x_2 - x_1) \left[\frac{1 - e^{-\alpha(n-1)}}{1 - e^{-\alpha(N-1)}} \right]$$

for concentration near the second end point. Here the strength of the concentration is controlled by the specified decay factor α , and N is the number of points on the curve.

CSPLØT (Plot)

The plot code CSPLØT plots the coordinate system generated by the code WESCØR, having read the coordinate system from file 11 as output by WESCØR. The input consists of the number of coordinate lines to be plotted, a designation for skipping lines, the extent of the field to be plotted, and a factor for using different seating in the horizontal and vertical directions. This input is detailed below.

The plot is formed by simply connecting the points on a line of constant curvilinear coordinates in the physical region, i.e., by constructing straight lines between each successive pair of points, $X(I,J)$ and $Y(I,J)$, as one index is held fixed.

WESCOR Input Instructions

```

100=C *****
110=C
120=C ***** W E S C O R *****
130=C
140=C *****
150=C
160=C 2-D BOUNDARY-FITTED COORDINATE SYSTEM CODE
170=C
180=C MISSISSIPPI STATE UNIVERSITY , 1982
190=C
200=C U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
210=C VICKSBURG, MISSISSIPPI
220=C
230=C *****
240=C
250=C ***** SLIT-SLAB CONFIGURATION *****
260=C *
270=C ***** ATTRACTION TO COORDINATE LINES/POINTS AND TO SPACE LINES/POINTS.
280=C ***** CONTROL FUNCTIONS ALSO INTERPOLATED FROM BOUNDARY POINT DISTRIBUTION.
290=C *
300=C *****
310=C ***** INPUT INSTRUCTIONS :
320=C *
330=C *** CARDS(2) : LABEL - FORMAT(10A8)
340=C *
350=C * LABEL - TWO 80 CHARACTER CARDS. (BLANK CARDS IF NO LABEL)
360=C *
370=C *** CARD : IMAX,JMAX,NBDY,ITER,ISLIT,IBNDY,IDISA,IWIR,IWINTL,
380=C * IWFIN,NREN - FORMAT(11I5)
390=C *
400=C * IMAX - NUMBER OF XI POINTS.
410=C *
420=C * JMAX - NUMBER OF ETA POINTS.
430=C *
440=C * NBDY - TOTAL NUMBER OF SLAB SIDES AND SLITS IN THE FIELD.
450=C *
460=C * ITER - MAXIMUM NUMBER OF ITERATIONS ALLOWED.
470=C *
480=C * ISLIT - =1 SLAB SIDES OR SLITS READ FROM CARDS.
490=C * X,Y - FORMAT(2F10.0) , ONE POINT PER CARD.
500=C * =2 SLAB SIDES OR SLITS READ FROM FILE 10.
510=C * X,Y - UNFORMATTED , ONE POINT PER IMAGE.
520=C *
530=C * (NOTE: HORIZONTAL SLITS ARE READ CLOCKWISE FROM RIGHT END.)
540=C * ( VERTICAL SLITS ARE COUNTER-CLOCKWISE FROM TOP. )
550=C * ( SLAB SIDES MAY BE READ IN EITHER DIRECTION. )
560=C *
570=C * IBNDY - =0 OUTER BOUNDARY CALCULATED INTERNALLY AS CIRCLE.
580=C * =1 OUTER BOUNDARY READ FROM CARDS.
590=C * X,Y - FORMAT(2F10.0) , ONE POINT PER CARD.
600=C * =2 OUTER BOUNDARY READ FROM FILE 10.
610=C * X,Y - UNFORMATTED , ONE POINT PER IMAGE.
620=C * =-1 OUTER BOUNDARY READ IN SEGMENTS AS SLAB SIDES.
630=C *
640=C * (NOTE: FOR IBNDY = 1 OR 2 , OUTER BOUNDARY IS READ CLOCKWISE)
650=C * ( FROM POINT (INF1,INFETA). )
660=C * ( "OUTER BOUNDARY" MEANS ENTIRE BOUNDARY OF TRANSFORMED )
670=C * ( REGION IF NREN=0. IF NREN IS NOT ZERO, THEN OUTER )
680=C * ( BOUNDARY IS THE TOP OF THE TRANSFORMED REGION AND )
690=C * ( INNER BOUNDARY IS THE BOTTOM. )
700=C *
710=C * IDISA - =0 DON'T READ OR WRITE SYSTEM FROM OR ON FILE.
720=C * =1 WRITE SYSTEM ON FILES 10 & 11. DON'T READ SYSTEM FROM FILE.
730=C * =2 WRITE SYSTEM ON FILES 10 & 11 , READ SYSTEM FROM FILE 10 FOR RESTART.
740=C * =3 READ SYSTEM FROM FILE 10 FOR RESTART. DON'T WRITE SYSTEM ON FILE 11.
750=C *
760=C * (NOTE: FILE 10 IS RESTART FILE FOR CONTINUATION OF ITERATION.)
770=C * ( FILE 11 IS STORAGE FILE FOR FINAL SYSTEM. )
780=C *
790=C * IWIR - =0 DON'T PRINT EACH ITERATION ERROR.
800=C * =1 PRINT EACH ITERATION ERROR.
810=C *
820=C * IWINTL - =0 DON'T PRINT INITIAL GUESS.
830=C * =1 PRINT INITIAL GUESS.
840=C *
850=C * IWFIN - NON-ZERO SUPPRESSES PRINT OF FINAL VALUES.

```

```

860=C  *
870=C  *
880=C  *      NREN - NON-ZERO USES RE-ENTRANT BOUNDARY ON LEFT & RIGHT SIDES
890=C  *      OF TRANSFORMED REGION, WITH OUTER BOUNDARY ON TOP
900=C  *      AND INNER BOUNDARY ON BOTTOM.
910=C  *
920=C  *      INNER BOUNDARY IS READ AS FOLLOWS BEFORE READING OUTER BOUNDARY:
930=C  *      =1 INNER BOUNDARY READ FROM CARDS.
940=C  *      X,Y - FORMAT(2F10.0), ONE POINT PER CARD.
950=C  *      =2 INNER BOUNDARY READ FROM FILE 10.
960=C  *      X,Y - UNFORMATTED, ONE IMAGE PER CARD.
970=C  *
980=C  *      (NOTE: SLITS AND/OR SLABS MAY ALSO BE PRESENT.)
990=C  *
1000=C *
1010=C *      *** CARDS(NBODY) LB1, LB2, LB3, LTYPE - FORMAT(4I5)
1020=C *
1030=C *      LB1, LB2 - FIRST AND LAST INDICES OF SLAB SIDE OR SLIT ENDS.
1040=C *      (LB2 MAY BE LESS THAN LB1 FOR SLAB SIDE. INPUT IS FROM LB1 TO LB2.)
1050=C *
1060=C *      LB3 - INDEX OF LINE ON WHICH SLAB SIDE OR SLIT IS LOCATED.
1070=C *
1080=C *      LTYPE - SLAB SIDE OR SLIT TYPE (1 FOR HORIZONTAL, 2 FOR VERTICAL.)
1090=C *
1100=C *      (NEGATIVE INDICATES SLAB SIDE, RATHER THAN SLIT.)
1110=C *      (SUBTRACT 10 FOR OUTER BOUNDARY SEGMENT.)
1120=C *      (I.E., -11 IS HORIZONTAL OUTER BOUNDARY SEGMENT.)
1130=C *      (-12 IS VERTICAL OUTER BOUNDARY SEGMENT.)
1140=C *
1150=C *      *** CARD R(1), R(2), R(3), YINFIN, AINFIN, XOINF, YOINF, INFXI, INFETA
1160=C *      - FORMAT(7F10.0, 2I5)
1170=C *
1180=C *      R(1) - SOR ACCELERATION PARAMETER.
1190=C *      (ZERO VALUE CAUSES VARIABLE ACCELERATION PARAMETER)
1200=C *      (FIELD TO BE CALCULATED INTERNALLY.)
1210=C *
1220=C *      R(2) - ALLOWABLE X ITERATION ERROR.
1230=C *
1240=C *      R(3) - ALLOWABLE Y ITERATION ERROR.
1250=C *
1260=C *      YINFIN - RADIUS OF CIRCULAR OUTER BOUNDARY.
1270=C *
1280=C *      AINFIN - ANGLE OF FIRST POINT ON CIRCULAR OUTER BOUNDARY (DEGREES).
1290=C *      (COUNTER-CLOCK FROM POSITIVE X-AXIS.)
1300=C *
1310=C *      XOINF, YOINF - CENTER OF CIRCULAR OUTER BOUNDARY.
1320=C *
1330=C *      NINF - NUMBER OF UNIQUE POINTS ON CIRCULAR OUTER BOUNDARY.
1340=C *
1350=C *      INFXI, INFETA - INDICES OF FIRST POINT ON CIRCULAR OUTER BOUNDARY.
1360=C *
1370=C *      (NOTE : LAST 7 OF THESE PARAMETERS ARE IRRELEVANT IF OUTER BOUNDARY IS READ.)
1380=C *
1390=C *
1400=C *      ** IF BODIES AND/OR OUTER BOUNDARY ARE READ FROM CARDS, SUCH CARDS
1410=C *      FOLLOW NEXT.
1420=C *
1430=C *      ** SLITS AND/OR SLAB SIDES ARE READ FIRST, THEN OUTER BOUNDARY IS READ.
1440=C *      (THESE RULES APPLY FOR READING FROM FILE 10 AS WELL AS FROM CARDS.)
1450=C *
1460=C *
1470=C *
1480=C *      ** IF NO COORDINATE ATTRACTION IS TO BE USED, FOLLOW THESE CARDS
1490=C *      WITH FIVE BLANK CARDS. IF ATTRACTION IS TO BE USED, USE THE FOLLOWING
1500=C *      INPUT RATHER THAN THE BLANK CARDS:
1510=C *
1520=C *      ** INPUT FOR COORDINATE SYSTEM CONTROL : USE FOUR SETS, ONE FOR
1530=C *      XI-LINE ATTRACTION TO COORDINATE LINES/POINTS, ONE FOR ETA-LINE ATTRACTION
1540=C *      TO COORDINATE LINES/POINTS, ONE FOR XI-LINE ATTRACTION TO SPACE LINES/POINTS,
1550=C *      AND ONE FOR ETA-LINE ATTRACTION TO SPACE LINES/POINTS.
1560=C *      ** ANY SET NOT WANTED IS REPLACED BY ONE BLANK CARD.
1570=C *
1580=C *      *****
1590=C *
1600=C *      ** THE FOLLOWING, MARKED WITH *, IS FOR ATTRACTION TO COORDINATE LINES/POINTS:
1610=C *
1620=C *      *** CARD : ATYP, ITYP, NLN, NPT, DEC, AMFFAC - FORMAT(A2, I2, 2I5, 2F10.0)
1630=C *
1640=C *      ATYP - TYPE OF ATTRACTION. (X1 FOR XI-LINE ATTRACTION,
1650=C *      ETA FOR ETA-LINE ATTRACTION.) LEFT JUSTIFIED.

```

```

1660=C ## ITYP - ZERO GIVES ATTRACTION ON BOTH SIDES.
1670=C ## NON-ZERO GIVES ATTRACTION ON UPPER SIDE AND
1680=C ## REPULSION ON LOWER SIDE.
1690=C ##
1700=C ## NLN - NUMBER OF ATTRACTION LINES.
1710=C ##
1720=C ## NPT - NUMBER OF ATTRACTION POINTS.
1730=C ##
1740=C ## DEC - NON-ZERO DEC USES DEC FOR DECAY FACTOR.
1750=C ##
1760=C ## AMPFAC - NON-ZERO AMPFAC MULTIPLIES ALL AMPLITUDES BY AMPFAC.
1770=C ##
1780=C ### CARDS(NLN) : JLN,ALN,DLN - FORMAT(5X,15,2F10.0)
1790=C ## (OMIT IF NLN IS ZERO)
1800=C ##
1810=C ## JLN - ATTRACTION LINE INDEX.
1820=C ##
1830=C ## ALN - AMPLITUDE (NEGATIVE REPELS) FOR LINE ATTRACTION.
1840=C ##
1850=C ## DLN - DECAY FACTOR FOR LINE ATTRACTION.
1860=C ##
1870=C ### CARDS(NPT) : IPT,JPT,APT,DPT - FORMAT(2I5,2F10.0)
1880=C ## (OMIT IF NPT IS ZERO)
1890=C ##
1900=C ## IPT,JPT - ATTRACTION POINT INDICES.
1910=C ##
1920=C ## APT - AMPLITUDE (NEGATIVE REPELS) FOR POINT ATTRACTION.
1930=C ##
1940=C ## DPT - DECAY FACTOR FOR POINT ATTRACTION.
1950=C ##
1960=C #####
1970=C ##
1980=C ### THE FOLLOWING, MARKED WITH $, IS FOR ATTRACTION TO SPACE LINES/POINTS :
1990=C ##
2000=C ### THE FOLLOWING CARDS ARE FOR ATTRACTION TO LINES AND/OR POINTS
2010=C ### DEFINED BY X,Y COORDINATES. IF NLN IS NOT ZERO, THEN NLN
2020=C ### OF THE CARDS GIVING NP MUST APPEAR. EACH OF THESE CARDS IS
2030=C ### FOLLOWED BY K OF THE CARDS GIVING XPT, ETC. IF NPT IS NOT
2040=C ### ZERO, THEN NPT OF THE CARDS GIVING XPT, ETC. MUST FOLLOW
2050=C ### THE LAST GROUP OF THESE CARDS.
2060=C ### ANY SET NOT WANTED IS REPLACED BY ONE BLANK CARD.
2070=C ##
2080=C ### CARD : ATYP,ITYP,NLN,NPT,DEC,AMPFAC - FORMAT(A8,I2,2I5,2F10.0)
2090=C ##
2100=C ## ATYP - TYPE OF ATTRACTION. (X1 FOR XI-LINE ATTRACTION,
2110=C ## ETA FOR ETA-LINE ATTRACTION.) LEFT JUSTIFIED.
2120=C ##
2130=C ## ITYP - ZERO GIVES ATTRACTION ON BOTH SIDES.
2140=C ## NON-ZERO GIVES ATTRACTION ON UPPER SIDE AND
2150=C ## REPULSION ON LOWER SIDE.
2160=C ##
2170=C ## NLN - NUMBER OF ATTRACTION LINES.
2180=C ##
2190=C ## NPT - NUMBER OF ATTRACTION POINTS.
2200=C ## (NOT INCLUDING POINTS ON ATTRACTION LINES)
2210=C ##
2220=C ## DEC - NON-ZERO DEC USES DEC FOR DECAY FACTOR.
2230=C ##
2240=C ## AMPFAC - NON-ZERO AMPFAC MULTIPLIES ALL AMPLITUDES BY AMPFAC.
2250=C ##
2260=C ### CARD : NP - FORMAT(I5)
2270=C ##
2280=C ## NP - NUMBER OF POINTS ON THIS ATTRACTION LINE.
2290=C ##
2300=C ### CARDS : XPT,YPT,APT,DPT,VEC1,VEC2 - FORMAT(6F10.0)
2310=C ##
2320=C ## XPT,YPT - COORDINATES OF ATTRACTION POINT OR
2330=C ## POINT ON ATTRACTION LINE.
2340=C ##
2350=C ## APT - ATTRACTION AMPLITUDE (NEGATIVE REPELS).
2360=C ##
2370=C ## DPT - DECAY FACTOR.
2380=C ##
2390=C ## VEC1,VEC2 - X,Y COMPONENTS OF UNIT VECTOR NORMAL TO
2400=C ## ATTRACTION DIRECTION FOR POINT ATTRACTION.
2410=C ## (CALCULATED INTERNALLY FOR LINE ATTRACTION.)
2420=C ##
2430=C ##
2440=C ## .....
2450=C ##

```



```

2460=C *** THE LAST COORDINATE SYSTEM CONTROL CARD IS THE FOLLOWING CARD :
2470=C *
2480=C *** CARD : IFAC,IRIT,EFAC - FORMAT(215,F10.0)
2490=C *
2500=C *      (CAN BE USED TO AID CONVERGENCE BY CONVERGING FIELD )
2510=C *      (WITH LESS ATTRACTION FIRST AND USING THIS RESULT )
2520=C *      (AS THE INITIAL GUESS FOR STRONGER ATTRACTION. )
2530=C *      (BLANK CARD MUST BE INPUT IF THIS FEATURE IS NOT USED.)
2540=C *      (STANDARD IS TO NOT USE THIS FEATURE , BUT ITS USE MAY)
2550=C *      (BE NECESSARY WITH STRONG ATTRACTION. )
2560=C *
2570=C *      IFAC - NUMBER OF STEPS IN ADDITION OF INHOMOGENEOUS TERM.
2580=C *      DOUBLES INHOMOGENEOUS TERM AT EACH STEP.
2590=C *
2600=C *      (ZERO CONVERGES WITH FULL ATTRACTION. )
2610=C *      (1.0 CONVERGES WITH NO ATTRACTION FIRST, THEN )
2620=C *      (WITH FULL ATTRACTION. 2.0 CONVERGES WITH NO )
2630=C *      (ATTRACTION FIRST, THEN WITH HALF, THEN WITH FULL.)
2640=C *      (INCREASE NUMBER OF STEPS IF DIVERGENCE OCCURS. )
2650=C *
2660=C *      IRIT - NON-ZERO VALUE CAUSES INHOMOGENEOUS TERM TO BE PRINTED.
2670=C *
2680=C *      EFAC - MULTIPLE OF CONVERGENCE CRITERION TO BE USED FOR
2690=C *      INTERMEDIATE CONVERGENCE BETWEEN ADDITIONS OF
2700=C *      INHOMOGENEOUS TERM. (TYPICALLY 10.0 )
2710=C *
2720=C *****
2730=C *
2740=C * MASS STORAGE FILES :
2750=C *
2760=C * RESTART FILE - FILE 10 :
2770=C *
2780=C *      (10) RXI,RETA
2790=C *      (10) X,Y,LSLIT,LABEL,IMAX,JMAX
2800=C *      (10) NBDY,NUMB,LE1,LE2,LE3,LTYPE,LFT,XL,XD,YL,YD,
2810=C *      NDI,NDI1,NDI2,NDI3,WACC
2820=C *
2830=C * COORDINATE SYSTEM STORAGE FILE - FILE 11 :
2840=C *
2850=C *      (11) LABEL,IMAX,JMAX
2860=C *      (11) ((LSLIT(I,J),I=1,IMAX,J=1,JMAX)
2870=C *      (11) ((X(I,J),I=1,IMAX,J=1,JMAX)
2880=C *      (11) ((Y(I,J),I=1,IMAX,J=1,JMAX)
2890=C *      (11) NBDY,NUMB,LE1,LE2,LE3,LTYPE,LFT,XL,XD,YL,YD,
2900=C *      NDI,NDI1,NDI2,NDI3
2910=C *
2920=C *****

```

LINES Input Instructions

```

100=C*****
110=C
120=C***** L I N E S *****
130=C
140=C*****
150=C
160=C BOUNDARY SEGMENT CODE FOR INPUT TO WESCOR
170=C
180=C MISSISSIPPI STATE UNIVERSITY . 1982
190=C
200=C U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
210=C VICKSBURG, MISSISSIPPI
220=C
230=C*****
240=C
250=C*** POINTS ON BOUNDARY SEGMENTS ***
260=C
270=C*****
280=C
290=C*** INPUT :
300=C
310=C**CARD : NLINES - FORMAT(I5)
320=C
330=C NLINES - TOTAL NUMBER OF LINES.
340=C
350=C**CARDS(NLINES) : N, ITYP, D1, D2, D3, D4, D5, D6, DE - FORMAT(2I5, 7F10.0)
360=C
370=C N - NUMBER OF POINTS ON LINE.
380=C
390=C ITYP - TYPE OF LINE
400=C 0 : STRAIGHT.
410=C 1 : CIRCULAR ARC.
420=C 2 : ELLIPTIC ARC.
430=C 3 : CURTC.
440=C 4 : QUADRATIC WITH ZERO SLOPE AT FIRST POINT.
450=C 5 : QUADRATIC WITH ZERO SLOPE AT SECOND POINT.
460=C
470=C D1-D6 AS FOLLOWS - ITEMS NOT CITED ARE IRRELEVANT)
480=C
490=C ITYP=0 : D1 - X OF FIRST POINT.
500=C D2 - Y OF FIRST POINT.
510=C D3 - X OF SECOND POINT.
520=C D4 - Y OF SECOND POINT.
530=C
540=C ITYP=1 : D1 - ANGLE OF FIRST POINT (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
550=C D2 - ANGLE OF SECOND POINT (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
560=C D3 - X OF CIRCLE CENTER.
570=C D4 - Y OF CIRCLE CENTER.
580=C D5 - CIRCLE RADIUS.
590=C
600=C ITYP=2 : D1 - ANGLE OF FIRST POINT. (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
610=C D2 - ANGLE OF SECOND POINT. (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
620=C D3 - X OF ELLIPSE CENTER.
630=C D4 - Y OF ELLIPSE CENTER.
640=C D5 - X-AXIS LENGTH OF ELLIPSE.
650=C D6 - Y-AXIS LENGTH OF ELLIPSE.
660=C
670=C ITYP=3 : D1-D4 SAME AS ITYP=0
680=C D5 - SLOPE AT FIRST POINT. (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
690=C D6 - SLOPE AT SECOND POINT. (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
700=C
710=C ITYP=4 : D1-D4 SAME AS ITYP=0
720=C
730=C ITYP=5 : D1-D4 SAME AS ITYP=0
740=C
750=C DE - EXPONENTIAL CONCENTRATION FACTOR.
760=C 0.0 FOR EQUAL SPACING ON LINE.
770=C NEGATIVE FOR CONCENTRATION NEAR FIRST POINT.
780=C POSITIVE FOR CONCENTRATION NEAR SECOND POINT.
790=C
800=C*****
810=C
820=C MASS STORAGE FILE :
830=C
840=C OUTPUT - FILE 10 : WRITE(10) X(I), Y(I)
850=C X & Y POINTS OF EACH LINE, INCLUDING E-DE.

```

CSPL0T Input Instructions

```

100=C*****
110=C#
120=C***** C S F L O T *****
130=C#
140=C*****
150=C#
160=C  COORDINATE SYSTEM PLOT CODE - MISSISSIPPI STATE UNIVERSITY , 1982
170=C
180=C                                U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
190=C                                VICKSBURG, MISSISSIPPI
200=C#
210=C*****
220=C
230=C *****
240=C #
250=C ***** INPUT INSTRUCTIONS :
260=C #
270=C *** CARD : NUMB , NUMB1 , ISKIP1 , ISKIP2 - FORMAT(4C)
280=C #
290=C #      NUMB - NUMBER OF ETA=CONSTANT LINES DESIRED FOR PLOT.
300=C #              (DEFAULT IS ALL LINES)
310=C #
320=C #      NUMB1 - NUMBER OF XI=CONSTANT LINES DESIRED FOR PLOT.
330=C #              (DEFAULT IS ALL LINES)
340=C #
350=C #      ISKIP1 - SKIP PARAMETER FOR XI=CONSTANT COORDINATE LINES.
360=C #              (1 PLOTS EVERY LINE, 2 PLOTS EVERY SECOND LINE, ETC.)
370=C #              (DEFAULT IS EVERY LINE)
380=C #
390=C #      ISKIP2 - SKIP PARAMETER FOR ETA=CONSTANT COORDINATE LINES.
400=C #              (SEE ISKIP1)
410=C #
420=C *** CARD : IB1 , IB2 , JB1 , JB2 - FORMAT(4I5)
430=C #
440=C #      I,J INDICES OF PLOT FIELD BOUNDARY.
450=C #      (I IS XI, J IS ETA.  DEFAULT IS ENTIRE FIELD)
460=C #
470=C *** CARD : XYRAT - FORMAT(710.0)
480=C #
490=C #      XYRAT - RATIO OF PLOTTED X TO Y LENGTHS.  (1.0)
500=C #
510=C *****
520=C #
530=C ***** COORDINATE SYSTEM IS READ UNFORMATTED FROM UNIT 10 AS
540=C ***** WRITTEN BY THE CODE 'WESCOR'.
550=C #
560=C *****

```

LINES Sample Runstream #1

```
120=JOB.
130=ACQUIRE,DN=BINARY,PIN=THOMPSONLINESB,ID=,DF=TR,UQ.
140=LDR,DN=BINARY.
150=DISPOSE,DN=FT10,SDN=FILE,ID=,DC=ST,DF=TR,TEXT=1
160='CATALOG,FILE,THOMPSONLINESB,ID=,RF=999.'.
170=DELETE,DN=BINARY.
180=EXIT.
190=DELETE,DN=BINARY.
200=$EOR
210=
220= 33 0 0.0 0.0 24.39 0.0
230= 9 0 0.0 -0.3 0.1 -0.3
240= 25 0 0.1 -0.3 24.39 -0.91
250= 25 0 0.0 -0.3 0.0 0.0
260= 25 0 24.39 -0.91 24.39 0.0
270=$EOR
280=$EOF
```

WESCOR Sample Runstream #1

```
120=JOB,T=60.
130=ACQUIRE,DN=FT10,PIN=THOMPSONLINESB,ID=,DF=TR,UQ.
140=ACQUIRE,DN=BINARY,PIN=THOMPSONCORDB,ID=,DF=TR,UQ.
150=LDR,DN=BINARY.
160=DISPOSE,DN=FT11,SDN=FILE,ID=,DC=ST,DF=TR,TEXT=1
170='CATALOG,FILE,THOMPSONCORDB,ID=,RF=999.'.
180=DELETE,DN=FT10.
190=DELETE,DN=BINARY.
200=EXIT.
210=DELETE,DN=FT10.
220=DELETE,DN=BINARY.
230=$EOR
240=JOHNSON FLUME
250=33 X 25 COORDINATE SYSTEM
260= 33 25 5 100 2 -1 1 1 1 0 0
270= 1 33 25 -11
280= 1 9 1 -11
290= 9 33 1 -11
300= 1 25 1 -12
310= 1 25 33 -12
320=1.8 0.00001 0.00001
330=
340=
350=
360=
370=
380=$EOR
390=$EOF
```

CSPL0T Sample Runstream

```
120=JOB.
130=ACQUIRE,DN=FT10,PIN=THOMPSONCORDB,ID=,DF=TR,UQ.
140=ACQUIRE,DN=BINARY,PIN=THOMPSONCSPL0TB,ID=,DF=TR,UQ.
150=LDR,LTB=METALIB,DN=BINARY.
160=DISPOSE,DN=FT01,SDN=FILE,ID=,DC=ST,DF=SR,TEXT=1
170='CATALOG,FILE,THOMPSONFLOT,ID=,RF=999.'.
180=DELETE,DN=FT10.
190=DELETE,DN=BINARY.
200=EXIT.
210=DELETE,DN=FT10.
220=DELETE,DN=BINARY.
230=$EOR
240= 0 0 1 1
250= 0 0 0 0
260=2.0
270=$EOR
280=$EOF
```

LINES Sample Runstream #2

```

120=JOB,
130=ACQUIRE,DN=BINARY,PON=THOMPSONLINES01,ID=,DF=TR,UQ.
140=LDR,DN=BINARY.
150=DISPOSE,DN=FT10,SEN=FILE,ID=,DC=ST,DF=TR,TEXT=1
160=CATALOG,FILE,THOMPSONLINES01,ID=,RF=999.
170=DELETE,DN=BINARY.
180=EXIT.
190=DELETE,DN=BINARY.
200=REOR
210= 15
220= 25 0 15.0 0.0 5.91 0.0 0.0 0.0 -0.08
230= 9 0 5.91 0.0 4.1 0.68 0.0 0.0 0.2
240= 3 0 4.1 0.68 3.9 0.68 0.0 0.0 0.0
250= 9 0 3.9 0.68 2.29 0.0 0.0 0.0 -0.2
260= 7 0 2.29 0.0 0.0 0.0 0.0 0.0 0.0
270= 23 0 0.0 0.0 0.0 1.63 0.0 0.0 0.0
280= 21 0 0.0 1.63 5.91 1.63 0.0 0.0 0.0
290= 29 0 5.91 1.63 15.0 1.63 0.0 0.0 0.04
300= 7 0 15.0 1.63 15.0 1.25 0.0 0.0 0.0
310= 5 3 15.0 1.25 15.25 0.96 -57.0 135.0 0.0
320= 5 0 15.25 0.96 15.8 0.43 0.0 0.0 0.0
330= 9 0 15.8 0.43 15.63 0.23 0.0 0.0 0.0
340= 5 0 15.63 0.23 15.25 0.58 0.0 0.0 0.0
350= 5 3 15.25 0.58 15.0 0.5 135.0 51.0 0.0
360= 9 0 15.0 0.5 15.0 0.0 0.0 0.0 0.0
370=REOR
380=REOF

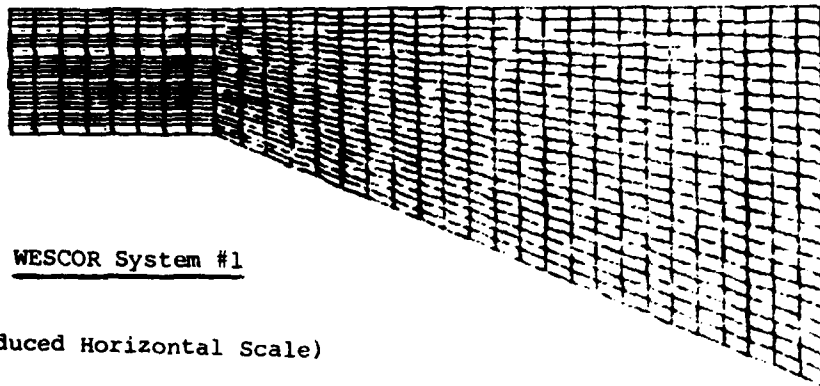
```

WESCØR Sample Runstream #2

```

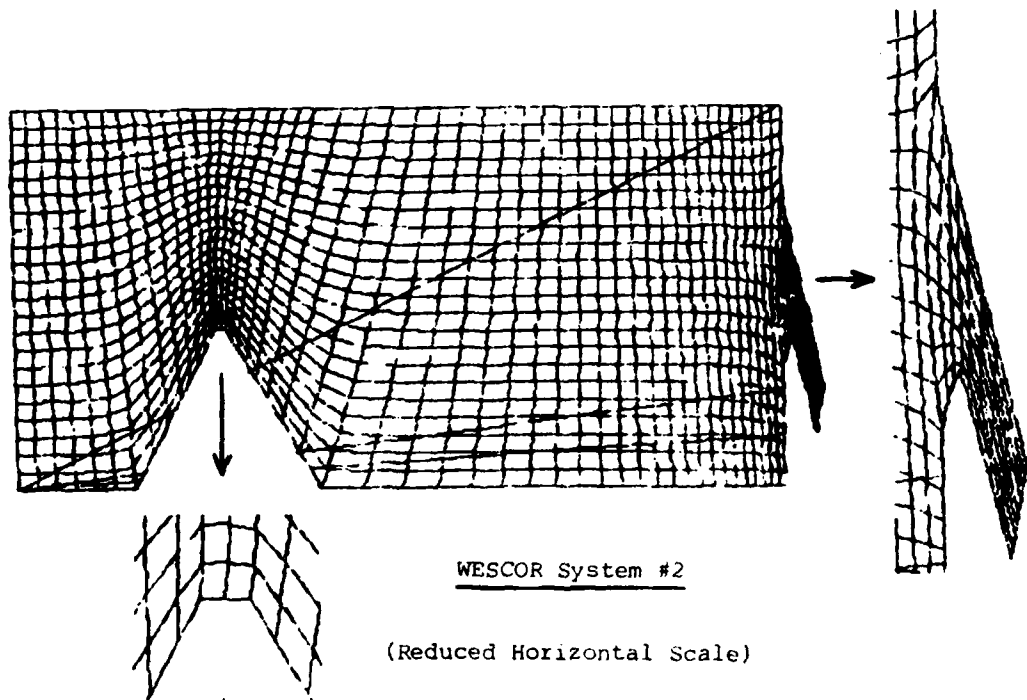
120=JOB,T=60.
130=ACQUIRE,DN=FT10,FIN=THOMPSONLINES01,ID=,DF=TR,UQ.
140=ACQUIRE,DN=BINARY,FON=THOMPSONCORD01,ID=,DF=TR,UQ.
150=LDR,DN=BINARY.
160=DISPOSE,DN=FT11,SEN=FILE,ID=,DC=ST,DF=TR,TEXT=1
170=CATALOG,FILE,THOMPSONCORD01,ID=,RF=999.
180=DELETE,DN=FT10.
190=DELETE,DN=BINARY.
200=EXIT.
210=DELETE,DN=FT10.
220=DELETE,DN=BINARY.
230=REOR
240=NORTON TEST 6 - WITH WFIR
250= 57 X 23 COORDINATE SYSTEM
260= 57 23 15 100 2 -1 1 1 1 0 0
270= 49 25 1 -11
280= 25 17 1 -11
290= 17 13 1 -11
300= 15 7 1 -11
310= 7 1 1 -11
320= 1 23 1 -12
330= 1 25 23 -11
340= 21 49 23 -11
350= 23 17 49 -2
360= 49 53 17 -1
370= 53 57 17 -1
380= 17 9 57 -12
390= 57 53 9 -1
400= 53 49 9 -1
410= 9 1 49 -2
420=1.8 0.00001 0.00001
430=
440=
450=
460=
470=
480=REOR
490=REOF

```



WESCOR System #1

(Reduced Horizontal Scale)



WESCOR System #2

(Reduced Horizontal Scale)

(Diagonal lines are spurious plotting errors)

FLUX-CORRECTED TRANSPORT IN AN EXPONENTIALLY STRETCHED GRID

Richard A. Schmalz, Jr.

U. S. Army Engineer Waterways Experiment Station
P. O. Box 631, Vicksburg, Mississippi 39180

ABSTRACT. The transport of a passive constituent in a tidal (oscillatory) flow regime is approximated using a Flux-Corrected Transport (FCT) technique over an exponentially stretched grid. The FCT technique consists of a nonlinear average of the results obtained from two schemes: a lower order in space nonoscillatory scheme, and a higher order in space scheme. The FCT method studied in this research employed a forward time upwind space (FTUS) scheme as the lower order non-oscillatory scheme and a forward time centered space (FTCS) scheme as the higher order scheme. Both schemes were implemented through an Alternating Direction Implicit (ADI) method employing the Thomas algorithm for matrix inversion.

Numerical results are obtained on a 6785 cell exponentially stretched grid employed to represent circulation in Mississippi Sound for the FTUS, FTCS, and FCT schemes. Comparison of these results demonstrates the magnitude of the differences for constituent transport obtained from these alternate numerical approaches on a real world problem size computational grid.

1. Introduction. The transport of a passive constituent (salinity) in a tidal (oscillatory) flow regime is considered using a Flux-Corrected Transport (FCT) technique in an exponentially stretched grid. The governing transport equation is first transformed using a C_1 piecewise exponential transformation. The development of the FCT approach for approximating the transformed equation is next presented. The FCT technique consists of a nonlinear average of the results obtained from two schemes: a lower order in space nonoscillatory scheme, and a higher order in space scheme. The FCT method studied in this research employs a forward time upwind space (FTUS) scheme as the lower order non-oscillatory scheme and a forward time centered space (FTCS) scheme as the higher order scheme. The implementation of these schemes through an Alternating Direction Implicit (ADI) method is then presented followed by a discussion of the limiting process within FCT.

The ADI multioperational FCT has been incorporated as a subroutine within the Waterways Experiment Station Implicit Flooding Model (WIFM). The revised model will be applied to Mississippi Sound and adjacent waters to study circulation and salinity patterns. Global effects are studied on a 6785 cell exponentially stretched grid. Numerical results for a sharp front problem over this global grid are presented for the FTUS, FTCS, and FCT schemes employing three different limiters. Results are compared to assess the magnitude of the differences on a real world

problem size computational grid. Conclusions and an outline for future work are finally presented.

2. TRANSPORT EQUATIONS IN CARTESIAN AND TRANSFORMED COORDINATES.

For tidal flow problems the following depth integrated form of the transport equation is initially considered.

$$(hs)_t + (hus)_x + (hvs)_y = \left[hK^x(s)_x \right]_x + \left[hK^y(s)_y \right]_y \quad (1)$$

where

- $h \equiv$ water depth
- $s \equiv$ constituent concentration
- $x \equiv$ x coordinate direction
- $y \equiv$ y coordinate direction
- $u \equiv$ velocity component in the x direction
- $v \equiv$ velocity component in the y direction
- $K^x \equiv$ effective dispersion coefficient in the x direction
- $K^y \equiv$ effective dispersion coefficient in the y direction
- $()_r \equiv \partial/\partial r$ for arbitrary variable r

The following exponential transformation has been considered by Wanstrath [1] and used by Butler in WIFM [2]. Consider an arbitrary variable x mapped or transformed into α . The transformation is carried out in a piecewise fashion by first partitioning x space as follows

$$(x_i^L, x_i^u) \quad i = 1, N \quad (2)$$

where

$x_i^L \equiv$ lower partition point for region i in x space

$x_i^u \equiv$ upper partition point for region i in x space

$N \equiv$ number of regions in the partition

Within each region of the partition the following transformation is considered.

$$x = a_i + b_i \alpha^{c_i} \quad x \in (x_i^L, x_i^u) \quad (3)$$

where a_i , b_i , c_i , and α are determined such that

$$\alpha \in (n_i^L \Delta \alpha, n_i^u \Delta \alpha) \quad n_i^L, n_i^u \in I, \quad n_i^u > n_i^L \quad (4a)$$

$$x_{i-1}^u = x_i^L \quad i = 2, \dots, N \quad (4b)$$

$$\left. \frac{dx}{da} \right|_{x_{i-1}^u} = \left. \frac{dx}{da} \right|_{x_i^L} \quad i = 2, \dots, N \quad (4c)$$

From Equations 4b and 4c, the transformation is C_1 .

Normally one specifies: x_i^L , $i = 1, \dots, N$, x_N^u , the partition points for the N regions, and $dx/da|_{x_i^L}$, $i = 1, \dots, N$, $dx/da|_{x_N^u}$, the rate of stretching at the partition points for the N regions. A time sharing code MAPIT [3] has been developed to compute n_i^u , a_i , b_i , c_i , $dx/da|_{x_i^u}^c$, $i = 1, \dots, N$. The computed rate of stretching $dx/da|_{x_i^u}^c$ is not exactly equal to the specified $dx/da|_{x_i^u}$, $i = 1, \dots, N$. However, the difference is usually negligible. Special options in the code allow the consideration of double regions; i.e., computations are performed simultaneously for two adjacent regions.

For the two-dimensional depth integrated transport equation, each coordinate is transformed separately; i.e., $x \rightarrow \alpha_1$ and $y \rightarrow \alpha_2$. Thus, we obtain

$$x = a_1^i + b_1^i c_1^i \quad x \in (x_i^L, x_i^u) \quad i = 1, N \quad (5a)$$

$$\alpha_1 \in (n_i^L \Delta \alpha_1, n_i^u \Delta \alpha_1)$$

$$y = a_2^i + b_2^i c_2^i \quad y \in (y_i^L, y_i^u) \quad i = 1, M \quad (5b)$$

$$\alpha_2 \in (m_i^L \Delta \alpha_2, m_i^u \Delta \alpha_2)$$

Let $\mu_1 = dx/da_1$ and $\mu_2 = dy/da_2$, then for an arbitrary variable P obtain:

$$(P)_x = \frac{(P)\alpha_1}{\mu_1} \quad (P)_y = \frac{(P)\alpha_2}{\mu_2} \quad (6)$$

Thus we may transform the depth integrated transport equation from $x - y$ to $\alpha_1 - \alpha_2$ space as follows:

$$(hs)_t + \frac{(hus)\alpha_1}{\mu_1} + \frac{(hvs)\alpha_2}{\mu_2} = \frac{1}{\mu_1} \left[hK^{\alpha_1} \frac{(s)\alpha_1}{\mu_1} \right]_{\alpha_1} + \frac{1}{\mu_2} \left[hK^{\alpha_2} \frac{(s)\alpha_2}{\mu_2} \right]_{\alpha_2} \quad (7)$$

This equation is the subject of numerical approximation. For convection much larger than diffusion as is the case in coastal flow regimes, the equation's character becomes predominantly hyperbolic making numerical approximation difficult.

3. FCT APPROACH. The sharp front problem has traditionally been employed to characterize the properties of finite difference approximations to the transport equation. Results of finite difference approximations for the sharp front problem fall into two general categories. Category I approximations produce severe oscillatory behavior for practical grid spacing but maintain the shape of the front. Category II approximations exhibit positively nonoscillatory solutions at the expense of frontal smearing. The Flux-Corrected Transport approach as developed by Boris and Book [4] and reformatted by Zalesak [5] attempts to obtain the best features from both categories; namely, a nonoscillatory solution to the sharp front problem with limited frontal smearing. In order to accomplish this, two schemes, one from each category, are considered. The schemes are written in the following flux format.

$$h_{n,m}^{k+1} S_{n,m}^I = h_{n,m}^k S_{n,m}^I - \left[\Delta\alpha_1(\mu_1) \Delta\alpha_2(\mu_2) \right]_n^{-1} \left(F_{n+1/2,m}^I - F_{n-1/2,m}^I + F_{n,m+1/2}^I - F_{n,m-1/2}^I \right) \quad (8)$$

such that $t = k\Delta t$, $x = \sum_i (\mu_1)_i \Delta\alpha_1$, $y = \sum_i (\mu_2)_i \Delta\alpha_2$

where $S_{n,m}^k \equiv$ concentration at location (n,m) at time level k

$h_{n,m}^k \equiv$ water depth at location (n,m) at time level k

$\Delta\alpha_1(\mu_1)_m \equiv$ x space step at m

$\Delta\alpha_2(\mu_2)_n \equiv$ y space step at n

$I \equiv$ General scheme index, which we set to H or L to designate the solution or flux at time level $k+1$ for the Category I or Category II scheme, respectively

$F_{n+1/2,m+1/2}^I \equiv$ Fluxes through the appropriate cell faces of cell (n,m) . Their form is dependent upon the FD scheme employed. H designates the higher order space fluxes; while L designates the lower order space fluxes

Note that the difference between the higher and lower order schemes may be written as follows:

$$\begin{aligned} (s_{n,m}^H - s_{n,m}^L) = & - \left[\Delta\alpha_1(\mu_1) \Delta\alpha_2(\mu_2) h_{n,m}^{k+1} \right]^{-1} \left(A_{n+1/2,m} - A_{n-1/2,m} \right. \\ & \left. + A_{n,m+1/2} - A_{n,m-1/2} \right) \end{aligned} \quad (9)$$

where anti-diffusive fluxes are given by:

$$A_{n+1/2,m+1/2} = F_{n+1/2,m+1/2}^H - F_{n+1/2,m+1/2}^L \quad (10)$$

These anti-diffusive fluxes are next limited, such that

$$A_{n+1/2,m+1/2}^C = C_{n+1/2,m+1/2} A_{n+1/2,m+1/2} \quad 0 < C_{n+1/2,m+1/2} < 1 \quad (11)$$

The crucial step in the Flux-Corrected Transport process is the determination of the limiting factors, $C_{n+1/2,m+1/2}$, which is deferred to a subsequent section.

Finally the updated FCT solution is determined as follows:

$$\begin{aligned} s_{n,m}^{k+1} = & s_{n,m}^L - \left[\Delta\alpha_1(\mu_1) \Delta\alpha_2(\mu_2) h_{n,m}^{k+1} \right]^{-1} \left(A_{n+1/2,m}^C - A_{n-1/2,m}^C \right. \\ & \left. + A_{n,m+1/2}^C - A_{n,m-1/2}^C \right) \end{aligned} \quad (12)$$

Note that if $C_{n+1/2,m} = 0 = C_{n,m+1/2}$, $s_{n,m}^{k+1} = s_{n,m}^L$ and for $C_{n+1/2,m} = 1.0 = C_{n,m+1/2}$, $s_{n,m}^{k+1} = s_{n,m}^H$.

Since, in general, $0 < C_{n+1/2,m+1/2} < 1$, $s_{n,m}^{k+1}$ is bounded by both $s_{n,m}^L$ and $s_{n,m}^H$.

4. FLUX-CORRECTED TRANSPORT IN A MULTIOPERATIONAL FORMAT. The transformed transport equation of Equation 7 is the subject of numerical approximation. A space staggered grid setup as shown in Figure 1 is employed. The datum convention is as shown in Figure 2.

Let us introduce the following notation as a prelude to the approximations. Define for an arbitrary variable $F_{n,m}^k$ where $t = k\Delta t$, $y = n\Delta y$, $x = m\Delta x$:

$$\delta_t^k(F_{n,m}^k) = F_{n,m}^{k+1/2} - F_{n,m}^k \quad (13a)$$

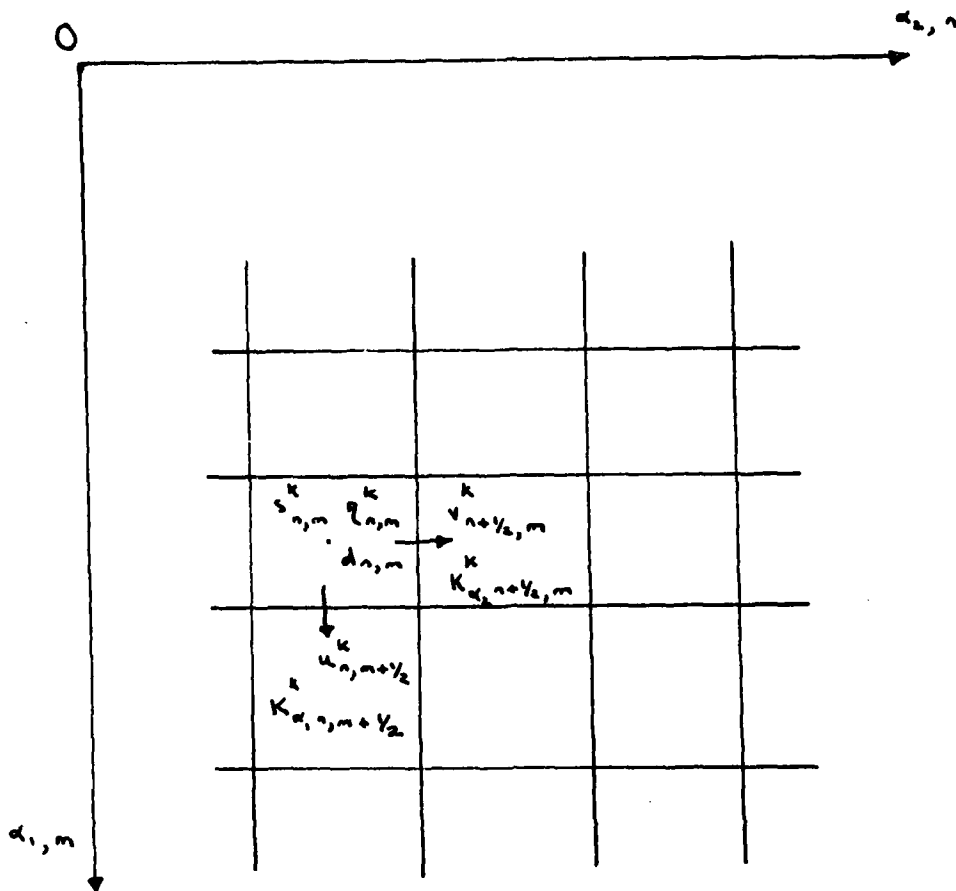


Figure 1. Space staggered finite difference grid in transformed coordinates

$$\delta_t^k(F_{n,m}^k) = F_{n,m}^{k+1} - F_{n,m}^k \quad (13b)$$

$$\delta_{\alpha_1}(F_{n,m}^k) = F_{n,m+1/2}^k - F_{n,m-1/2}^k \quad (13c)$$

$$\delta_{\alpha_2}(F_{n,m}^k) = F_{n+1/2,m}^k - F_{n-1/2,m}^k \quad (13d)$$

$$\frac{\alpha_1}{F_{n,m}} = \frac{(F_{n,m+1/2}^k + F_{n,m-1/2}^k)}{2} \quad (13e)$$

$$\frac{\alpha_2}{F_{n,m}} = \frac{(F_{n+1/2,m}^k + F_{n-1/2,m}^k)}{2} \quad (13f)$$

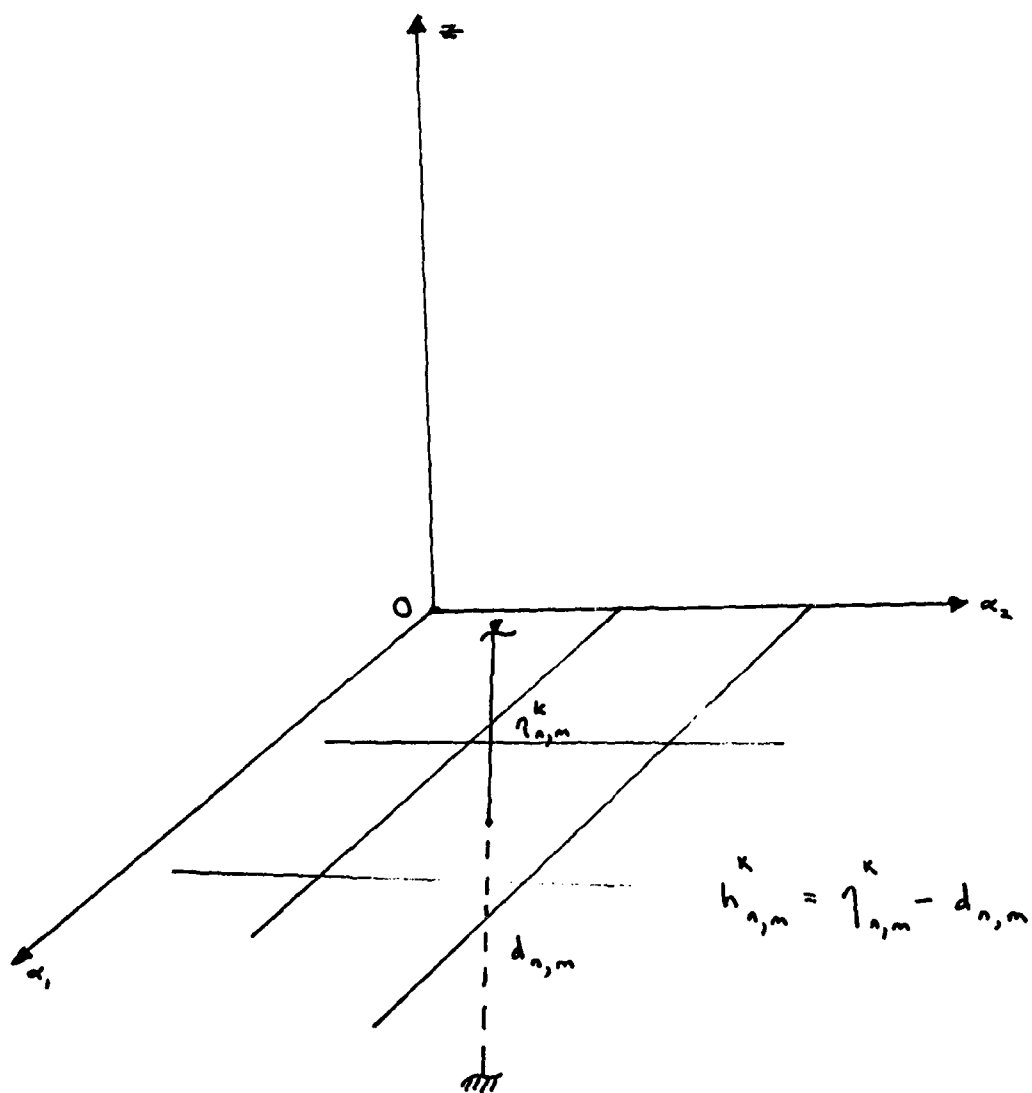


Figure 2. Datum convention employed within the space staggered grid system

AD-A118 920

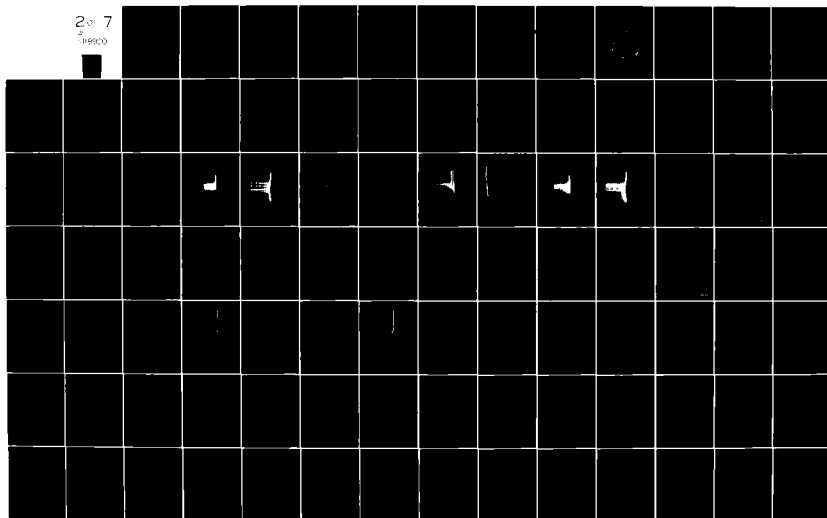
ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82

F/O 12/1

UNCLASSIFIED ARO-82-3

NL

23-7
189900



and the following upwind difference operations dependent upon an arbitrary auxiliary field variable, $f_{n,m}^k$.

$$\frac{f^k}{F_1} = \begin{cases} F_{n,m-1/2}^k & f_{n,m}^k \geq 0 \\ F_{n,m+1/2}^k & f_{n,m}^k < 0 \end{cases} \quad (14a)$$

$$\frac{f^k}{F_2} = \begin{cases} F_{n-1/2,m}^k & f_{n,m}^k \geq 0 \\ F_{n+1/2,m}^k & f_{n,m}^k < 0 \end{cases} \quad (14b)$$

Within the FCT approach, two solution schemes are considered. The standard Crank-Nicolson and their multioperational counterpart are presented in turn for the Category I FTCS and Category II FTUS schemes.

5. FTCS CATEGORY I SCHEME. The following finite difference equation is considered as an approximation to the transformed transport equation:

$$\begin{aligned} \delta_t^k(hs) + \frac{\Delta t}{2\Delta\alpha_1(\mu_1)_m} \delta_{\alpha_1} \left(\frac{\alpha_1}{h^{k+1}} \frac{\alpha_1}{s^{k+1}} u^{k+1} + \frac{\alpha_1}{h^k} \frac{\alpha_1}{s^k} u^k \right) \\ + \frac{\Delta t}{2\Delta\alpha_2(\mu_2)_n} \delta_{\alpha_2} \left(\frac{\alpha_2}{h^{k+1}} \frac{\alpha_2}{s^{k+1}} v^{k+1} + \frac{\alpha_2}{h^k} \frac{\alpha_2}{s^k} v^k \right) \\ - \frac{\Delta t}{2(\Delta\alpha_1)^2(\mu_1)_m} \delta_{\alpha_1} \left[\frac{\alpha_1}{h^{k+1}} \frac{\alpha_1}{s^{k+1}} \frac{\delta_{\alpha_1}(s^{k+1})}{(\mu_1)_m} + \frac{\alpha_1}{h^k} \frac{\alpha_1}{s^k} \frac{\delta_{\alpha_1}(s^k)}{(\mu_1)_m} \right] \\ - \frac{\Delta t}{2(\Delta\alpha_2)^2(\mu_2)_n} \delta_{\alpha_2} \left[\frac{\alpha_2}{h^{k+1}} \frac{\alpha_2}{s^{k+1}} \frac{\delta_{\alpha_2}(s^{k+1})}{(\mu_2)_n} \right. \\ \left. + \frac{\alpha_2}{h^k} \frac{\alpha_2}{s^k} \frac{\delta_{\alpha_2}(s^k)}{(\mu_2)_n} \right] = 0 \quad \text{at } (n,m) \end{aligned} \quad (15)$$

The solution of the above semi-implicit difference scheme requires the inversion of a large unbanded matrix. In order to reduce computational effort, the following ADI multioperational difference equations are used.

The approximations for the α_1 -X-Sweep are given as follows:

$$\begin{aligned}
 \delta_t^k (hs) + \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1(\mu_1)_m} & \left(\frac{\alpha_1}{h^{k+1/2}} \frac{\alpha_1}{s^{k+1/2}} \frac{\alpha_1}{u^{k+1/2}} \right) \\
 - \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1^2(\mu_1)_m} & \left[\frac{\alpha_1}{h^{k+1/2}} K_{\alpha_1}^{k+1/2} \frac{\delta_{\alpha_1}(s)^{k+1/2}}{(\mu_1)_m} \right] \\
 + \frac{\Delta t}{2(\mu_2)_n \Delta\alpha_2} \delta_{\alpha_2} & \left(\frac{\alpha_2}{h^k} \frac{\alpha_2}{s^k} \frac{\alpha_2}{v^k} \right) \\
 - \frac{\Delta t \delta_{\alpha_2}}{2\Delta\alpha_2^2(\mu_2)_n} & \left[\frac{\alpha_2}{h^k} K_{\alpha_2}^k \frac{\delta_{\alpha_2}(s^k)}{(\mu_2)_n} \right] = 0 \text{ at } (n,m)
 \end{aligned} \tag{16a}$$

The approximations for the α_2 -Y-Sweep are as follows:

$$\begin{aligned}
 \delta_t^{k+1/2} (hs) + \frac{\Delta t \delta_{\alpha_2}}{2\Delta\alpha_2(\mu_2)_n} & \left(\frac{\alpha_2}{h^{k+1}} \frac{\alpha_2}{s^{k+1}} \frac{\alpha_2}{v^{k+1}} \right) \\
 - \frac{\Delta t \delta_{\alpha_2}}{2\Delta\alpha_2^2(\mu_2)_n} & \left[\frac{\alpha_2}{h^{k+1}} K_{\alpha_2}^{k+1} \frac{\delta_{\alpha_2}(s^{k+1})}{(\mu_2)_n} \right] \\
 + \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1(\mu_1)_m} & \left(\frac{\alpha_1}{h^{k+1/2}} \frac{\alpha_1}{s^{k+1/2}} \frac{\alpha_1}{u^{k+1/2}} \right) \\
 - \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1^2(\mu_1)_m} & \left[\frac{\alpha_1}{h^{k+1/2}} K_{\alpha_1}^{k+1/2} \frac{\delta_{\alpha_1}(s^{k+1/2})}{(\mu_1)_m} \right] = 0 \text{ at } (n,m)
 \end{aligned} \tag{16b}$$

The Thomas algorithm may be used to invert the tridiagonal matrices in each sweep with minimal computational effort.

6. FTUS CATEGORY II SCHEME. The following finite difference equation is considered as an approximation to the transformed transport equation:

$$\begin{aligned}
 & \delta_t^k (hs) + \frac{\Delta t}{2\Delta\alpha_1(\mu_1)_m} \delta_{\alpha_1} \left(\frac{\alpha_{1,k+1} u^{k+1}}{h^{k+1} s_1} u^{k+1} + \frac{\alpha_{1,k} u^k}{h^k s_1} u^k \right) \\
 & + \frac{\Delta t}{2\Delta\alpha_2(\mu_2)_n} \delta_{\alpha_2} \left(\frac{\alpha_{2,k+1} v^{k+1}}{h^{k+1} s_2} v^{k+1} + \frac{\alpha_{2,k} v^k}{h^k s_2} v^k \right) \\
 & - \frac{\Delta t}{2(\Delta\alpha_1)^2(\mu_1)_m} \delta_{\alpha_1} \left[\frac{\alpha_{1,k+1} K_{\alpha_1}^{k+1}}{h^{k+1}} \frac{\delta_{\alpha_1}(s^{k+1})}{(\mu_1)_m} + \frac{\alpha_{1,k} K_{\alpha_1}^k}{h^k} \frac{\delta_{\alpha_1}(s^k)}{(\mu_1)_m} \right] \\
 & - \frac{\Delta t}{2(\Delta\alpha_2)^2(\mu_2)_n} \delta_{\alpha_2} \left[\frac{\alpha_{2,k+1} K_{\alpha_2}^{k+1}}{h^{k+1}} \frac{\delta_{\alpha_2}(s^{k+1})}{(\mu_2)_n} \right. \\
 & \quad \left. + \frac{\alpha_{2,k} K_{\alpha_2}^k}{h^k} \frac{\delta_{\alpha_2}(s^k)}{(\mu_2)_n} \right] = 0 \text{ at } (n,m)
 \end{aligned} \tag{17}$$

To affect the solution of this scheme again the inversion of an unbanded matrix is required. To reduce computational effort, the following ADI multioperational difference equations are utilized.

α_1 -X Sweep:

$$\begin{aligned}
 & \delta_t^k (hs) + \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1(\mu_1)_m} \left(\frac{\alpha_1}{h^{k+1/2}} \frac{u^{k+1/2}}{s^{k+1/2}} u^{k+1/2} \right) \\
 & - \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1^2(\mu_1)_m} \left[\frac{\alpha_1}{h^{k+1/2}} K_{\alpha_1}^{k+1/2} \frac{\delta_{\alpha_1}(s^{k+1/2})}{(\mu_1)_m} \right] \\
 & + \frac{\Delta t}{2(\mu_2)_n} \delta_{\alpha_2} \left(\frac{\alpha_2 v^k}{h^k s^k v^k} \right) \\
 & - \frac{\Delta t \delta_{\alpha_2}}{2\Delta\alpha_2^2(\mu_2)_n} \left[\frac{\alpha_1}{h^k K_{\alpha_2}^k} \frac{\delta_{\alpha_2}(s^k)}{(\mu_2)_n} \right] = 0 \text{ at } (n,m)
 \end{aligned} \tag{18a}$$

α_2 -Y Sweep:

$$\begin{aligned}
 & \delta_t^{k+1/2} (hs) + \frac{\Delta t \delta_{\alpha_2}}{2\Delta\alpha_2(\mu_2)_n} \left(\frac{\alpha_2}{h^{k+1}} \frac{v^{k+1}}{s^{k+1}} v^{k+1} \right) \\
 & - \frac{\Delta t \delta_{\alpha_2}}{2\Delta\alpha_2^2(\mu_2)_n} \left[\frac{\alpha_2}{h^{k+1}} K_{\alpha_2}^{k+1} \frac{\delta_{\alpha_2}(s^{k+1})}{(\mu_2)_n} \right] \\
 & + \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1(\mu_1)_m} \left(\frac{\alpha_1}{h^{k+1/2}} \frac{u^{k+1/2}}{s^{k+1/2}} u^{k+1/2} \right) \\
 & - \frac{\Delta t \delta_{\alpha_1}}{2\Delta\alpha_1^2(\mu_1)_m} \left[\frac{\alpha_1}{h^{k+1/2}} K_{\alpha_1}^{k+1/2} \frac{\delta_{\alpha_1}(s^{k+1/2})}{(\mu_1)_m} \right] = 0 \text{ at } (n,m)
 \end{aligned} \tag{18b}$$

The standard Crank-Nicolson equations are assumed to be contained within their corresponding multioperational difference equations. For the linear case obtained for $(\mu_2)_n = (\mu_1)_n = 1$, K^{a1} , K^{a2} , u , v , and h constant in space and time, the intermediate time level may be eliminated in the multioperational approach and the total difference equation obtained equals the standard difference equation plus some higher order in time factorization terms. The total difference equation is also consistent with the linear transport equation. For the nonlinear case considered, it is not possible to eliminate the constituent intermediate time level. Thus the exact form of the factorization terms may not be determined. However, their numerical effect is small and may be tested in the following manner.

Consider the standard Crank-Nicolson approximations, $\zeta_{CN}^I \left[\begin{pmatrix} S_{CN}^{k+1} \\ S_{n,m}^k \end{pmatrix} \right]$, where $I = H$ for FTCS, and $I = L$ for FTUS. Similarly, denote the corresponding multioperational ADI approximations, $\zeta_{ADI}^I \left[\begin{pmatrix} S_{ADI}^{k+1} \\ S_{n,m}^k \end{pmatrix} \right]$, where $I = H$ for FTCS and L for FTUS.

First compute, implicitly,

$$\left(S_{ADI}^{k+1} \right)_{n,m}^I = \zeta_{ADI}^I \left[\begin{pmatrix} S_{ADI}^{k+1} \\ S_{n,m}^k \end{pmatrix} \right]_{I=H \text{ and } L} \quad (19)$$

then compute, explicitly,

$$\left(S_{CN'}^{k+1} \right)_{n,m}^I = \zeta_{CN}^I \left[\begin{pmatrix} S_{ADI}^{k+1} \\ S_{n,m}^k \end{pmatrix} \right]_{I=H \text{ and } L} \quad (20)$$

Compare $\left(S_{ADI}^{k+1} \right)_{n,m}^I$ and $\left(S_{CN'}^{k+1} \right)_{n,m}^I$ for $I = H$ and L . Numerical tests confirm that the two solutions are practically identical, thus

$$\left(S_{ADI}^{k+1} \right)_{n,m}^I \approx \left(S_{CN'}^{k+1} \right)_{n,m}^I \approx \left(S_{CN}^{k+1} \right)_{n,m}^I$$

The standard Crank-Nicolson equations may be written in the flux form of Equation 8; thus they by their nature are mass conservative. Thus the multioperational schemes are also mass conservative. To implement FCT,

$\left(S_{ADI}^{k+1} \right)_{n,m}^I$ are computed, then $\zeta_{CN}^I \left[\begin{pmatrix} S_{ADI}^{k+1} \\ S_{n,m}^k \end{pmatrix} \right]$ is employed to compute the fluxes $F_{n+1/2, m+1/2}^I$ shown in Equation 8. Computations outlined in Equations 9-11 are next performed. In Equation 12, $S_{n,m}^L = \left(S_{ADI}^{k+1} \right)_{n,m}^L$.

7. FLUX-CORRECTED TRANSPORT LIMITERS. The crucial step in the FCT method is the limiting of the anti-diffusive fluxes; e.g., the determination of $C_{n+1/2,m+1/2}$ in Equation 11. Zalesak employs the following method [5]. In what follows, $S_{n,m}^L = (S_{ADI}^{k+1})_{n,m}^L$.

The anti-diffusive fluxes are first screened as follows:

$$\begin{aligned} A_{n+1/2,m} &= 0 & \text{if } A_{n+1/2,m} (S_{n+1,m}^L - S_{n,m}^L) < 0 \\ & & \text{and either } A_{n+1/2,m} (S_{n+2,m}^L - S_{n+1,m}^L) < 0 \quad (21a) \\ & & \text{or } A_{n+1/2,m} (S_{n,m}^L - S_{n-1,m}^L) < 0 \end{aligned}$$

$$\begin{aligned} A_{n,m+1/2} &= 0 & \text{if } A_{n,m+1/2} (S_{n,m+1}^L - S_{n,m}^L) < 0 \\ & & \text{and either } A_{n,m+1/2} (S_{n,m+2}^L - S_{n,m+1}^L) < 0 \quad (21b) \\ & & \text{or } A_{n,m+1/2} (S_{n,m}^L - S_{n,m-1}^L) < 0 \end{aligned}$$

Cell maximum and minimum values are computed.

$$S_{n,m}^{a'} = \max(S_{n,m}^k, S_{n,m}^L) \quad S_{n,m}^{b'} = \min(S_{n,m}^k, S_{n,m}^L) \quad (22a)$$

$$S_{n,m}^{\max} = \max(S_{n-1,m}^{a'}, S_{n,m}^{a'}, S_{n+1,m}^{a'}, S_{n,m-1}^{a'}, S_{n,m+1}^{a'}) \quad (22b)$$

$$S_{n,m}^{\min} = \min(S_{n-1,m}^{b'}, S_{n,m}^{b'}, S_{n+1,m}^{b'}, S_{n,m-1}^{b'}, S_{n,m+1}^{b'}) \quad (22c)$$

Next the sum of all anti-diffusive fluxes into and out of cell (n,m), $P_{n,m}^+$ and $P_{n,m}^-$, respectively, are determined.

$$\begin{aligned} P_{n,m}^+ &= \max(0, A_{n-1/2,m}) - \min(0, A_{n+1/2,m}) + \max(0, A_{n,m-1/2}) \\ & \quad - \min(0, A_{n,m+1/2}) \end{aligned} \quad (23)$$

$$\begin{aligned} P_{n,m}^- &= \max(0, A_{n+1/2,m}) - \min(0, A_{n-1/2,m}) + \max(0, A_{n,m+1/2}) \\ & \quad - \min(0, A_{n,m-1/2}) \end{aligned} \quad (24)$$

The maximum allowable mass into cell (n,m), $Q_{n,m}^+$, such that

$S_{n,m}^{k+1} \leq S_{n,m}^{\max}$ and the maximum allowable mass out of cell (n,m), $Q_{n,m}^-$, such that $S_{n,m}^{k+1} \geq S_{n,m}^{\min}$ are computed as follows:

$$Q_{n,m}^+ = (S_{n,m}^{\max} - S_{n,m}^L) \left[(\mu_1)_m \Delta \alpha_1 (\mu_2)_n \Delta \alpha_2 h_{n,m}^{k+1} \right] \quad (25)$$

$$Q_{n,m}^- = (S_{n,m}^L - S_{n,m}^{\min}) \left[(\mu_1)_m \Delta \alpha_1 (\mu_2)_n \Delta \alpha_2 h_{n,m}^{k+1} \right] \quad (26)$$

The following ratios are next calculated for use in determining the limiting coefficients.

$$R_{n,m}^+ = \begin{cases} \min(1, Q_{n,m}^+ / P_{n,m}^+) & P_{n,m}^+ > 0 \\ 0 & P_{n,m}^+ = 0 \end{cases} \quad (27)$$

$$R_{n,m}^- = \begin{cases} \min(1, Q_{n,m}^- / P_{n,m}^-) & P_{n,m}^- > 0 \\ 0 & P_{n,m}^- = 0 \end{cases} \quad (28)$$

The limiting coefficients are then given by

$$C_{n+1/2,m} = \begin{cases} \min(R_{n+1,m}^+, R_{n,m}^-) & A_{n+1/2,m} \geq 0 \\ \min(R_{n,m}^+, R_{n+1,m}^-) & A_{n+1/2,m} < 0 \end{cases} \quad (29)$$

$$C_{n,m+1/2} = \begin{cases} \min(R_{n,m+1}^+, R_{n,m}^-) & A_{n,m+1/2} \geq 0 \\ \min(R_{n,m}^+, R_{n,m+1}^-) & A_{n,m+1/2} < 0 \end{cases} \quad (30)$$

The author has considered two alternative limiters. In alternative one, Equation 22 is replaced with the following expressions

$$S_{n,m}^{\max} = \max(S_{n-1,m}^k, S_{n,m}^k, S_{n+1,m}^k, S_{n,m-1}^k, S_{n,m+1}^k) \quad (22a)'$$

$$S_{n,m}^{\min} = \min(S_{n-1,m}^k, S_{n,m}^k, S_{n+1,m}^k, S_{n,m-1}^k, S_{n,m+1}^k) \quad (22b)'$$

This alternative is designated as a mixed time level limiter, since $S_{n,m}^{\max}$ and $S_{n,m}^{\min}$ depend only on time level k values, while Q^+ and Q^- depend upon time level $k+1$ values of the lower order scheme. To obtain the second alternative limiter dependent only on previous time level k , employ Equation 22' and consider the following relations for Q^+ and Q^- :

$$Q_{n,m}^+ = (S_{n,m}^{\max} - S_{n,m}^k) \left[(\mu_1)_m \Delta \alpha_1 (\mu_2)_n \Delta \alpha_2 h_{n,m}^{k+1} \right] \quad (25)'$$

$$Q_{n,m}^- = (S_{n,m}^k - S_{n,m}^{\min}) \left[(\mu_1)_m \Delta \alpha_1 (\mu_2)_n \Delta \alpha_2 h_{n,m}^{k+1} \right] \quad (26)'$$

8. APPLICATION TO MISSISSIPPI SOUND. The Flux-Corrected Transport algorithm was incorporated as a group of subroutines in the Waterways Implicit Flooding Model (WIFM). Density coupling was not considered. Since the Mississippi Sound Project is concerned only with fixed boundary problems, the flooding routine was removed from WIFM. The resulting hydrodynamic-salinity code enables the treatment of the general nonlinear problem on a variably stretched grid. It is this problem which is the focus of the Mississippi Sound Numerical Investigation.

An exponentially stretched grid has been developed as shown in Figure 3 to describe global circulation and horizontal salinity variation. The grid employs 6785 computational cells with a minimum spatial resolution of approximately 4000 feet corresponding to areas within the passes between the barrier islands. Maximum grid spacing is achieved on the bottom and right hand side of the grid. This spacing of 15' latitude and longitude enables the grid to link directly to a 15' latitude and longitude grid employed to compute Gulf of Mexico tidal variations. The flexibility of the exponential transformation to provide areas of high resolution and spacing compatible with coarser grids for boundary driving is demonstrated.

In order to describe the dispersion mechanics within Mississippi Sound the following relations will be considered [6].

$$K^{\alpha_1} = D\sqrt{g} \frac{|u|h}{C} + R^{\alpha_1} \quad (31a)$$

$$K^{\alpha_2} = D\sqrt{g} \frac{|v|h}{C} + R^{\alpha_2} \quad (31b)$$

where

$K^{\alpha_1}, K^{\alpha_2} \equiv$ dispersion coefficients (ft^2/sec)
 $D \equiv$ dimensionless constant dependent upon the local flow

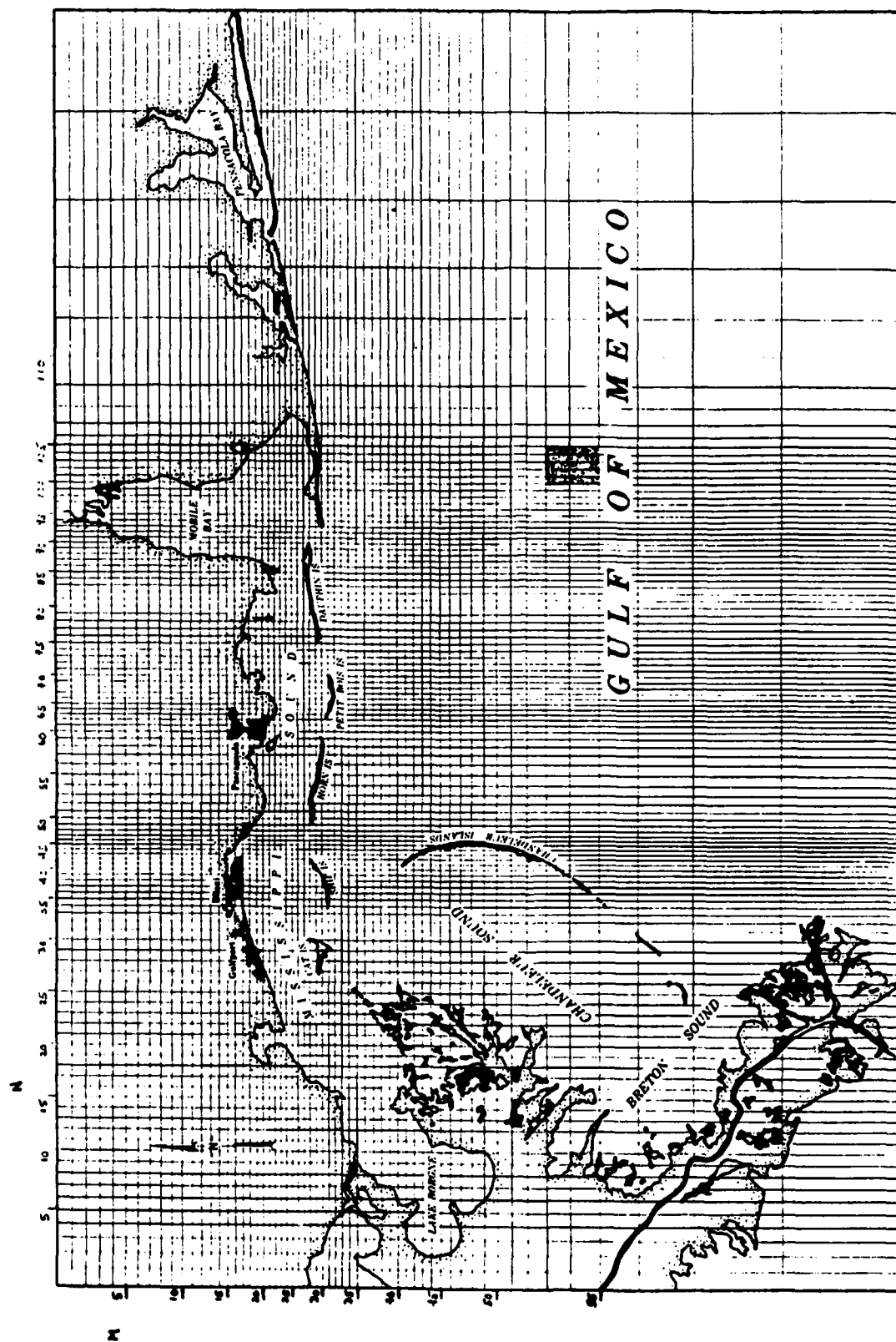


Figure 3. Exponentially stretched Mississippi Sound global grid

conditions [(5.93 - 20.2) in the direction of flow
0.23 in the direction perpendicular to flow]

$C \equiv$ Chezy coefficient ($\text{ft}^{1/2}/\text{sec}$)

$g \equiv$ gravity (ft/sec^2)

$h \equiv$ water depth (ft)

$u, v \equiv$ velocity components in the $x-\alpha_1$ and $y-\alpha_2$ directions,
respectively (ft/sec)

$R^{\alpha_1}, R^{\alpha_2} \equiv$ additional dispersion effects (ft^2/sec)

9. SHARP FRONT PROBLEM. The following sharp front problem was employed to test the FCT method for application to Mississippi Sound.

The hydrodynamic problem setup employed initial water surface elevations set to zero over the computational domain, flow inputs of $4 \text{ ft}^2/\text{sec}$ per unit flow width at cells (97,3) and (2,44), and a ramp function of $1/6'$ per time step as the seaward boundary elevation condition.

The salinity problem setup employed initial zero levels over the entire computational region except in a block of ten cells as shown below and indicated as the shaded area in Figure 3.

$$S_{n,m}^0 = \begin{cases} 0.0 & n \notin (101,105) \\ & m \notin (54,55) \\ 10.0 & n \in (101,105) \\ & m \in (54,55) \end{cases} \quad (32)$$

Salinity levels were maintained at zero at the two flow input locations.

In the dispersion relations, $D = 10.0$ and $R^{\alpha_1} = R^{\alpha_2} = 0.0 \text{ ft}^2/\text{sec}$.

A 5 time step (5τ) simulation was performed for several numerical schemes employing a time step length of 6 minutes resulting in a maximum gravity wave speed Courant number of 4 within Mississippi Sound.

In order to characterize the transport aspects of these simulations the following dimensionless numbers are computed.

$$Cr_{x,n,m}^k = \frac{|u_{n,m+1/2}^k| \Delta t}{(\mu_1)_m \Delta \alpha_1} \quad (33a)$$

$$Cr_{y,n,m}^k = \frac{|v_{n+1/2,m}^k| \Delta t}{(\mu_2)_n \Delta \alpha_2} \quad (33b)$$

$$Pe_{x,n,m}^k = \frac{|u_{n,m+1/2}^k| (\mu_1)_m \Delta \alpha_1}{\alpha_1^k K_{n,m+1/2}} \quad (34a)$$

$$Pe_{y,n,m}^k = \frac{|v_{n+1/2,m}^k| (\mu_2)_n \Delta \alpha_2}{\alpha_2^k K_{n+1/2,m}} \quad (34b)$$

where

- $Cr_{x,n,m}^k \equiv x - \alpha_1$ cell (n,m) Courant transport number at time k
- $Cr_{y,n,m}^k \equiv y - \alpha_2$ cell (n,m) Courant transport number at time k
- $Pe_{x,n,m}^k \equiv x - \alpha_1$ cell (n,m) Peclet number at time k
- $Pe_{y,n,m}^k \equiv y - \alpha_2$ cell (n,m) Peclet number at time k
- $\Delta t \equiv$ time step length
- $\Delta \alpha_1 \equiv \alpha_1$ space increment
- $\Delta \alpha_2 \equiv \alpha_2$ space increment
- $(\mu_1)_m \equiv$ cell (n,m) stretching coefficient in the α_1 direction
- $(\mu_2)_n \equiv$ cell (n,m) stretching coefficient in the α_2 direction
- $u_{n,m+1/2}^k \equiv$ velocity component in cell (n,m) at time k in the α_1 direction
- $v_{n+1/2,m}^k \equiv$ velocity component in cell (n,m) at time k in the α_2 direction
- $\alpha_1^k K_{n,m+1/2} \equiv$ dispersion coefficient in cell (n,m) at time k in the α_1 direction
- $\alpha_2^k K_{n+1/2,m} \equiv$ dispersion coefficient in cell (n,m) at time k in the α_2 direction

In general, these dimensionless numbers vary in time as well as in space over the computational grid and time interval of concern. Normal practice is to replace the time dependency of the cell velocity components by their maximum values, thus removing the time dependency. In all simulations, the following relations for the transport cell Courant numbers hold.

$$Cr_{x,n,m}^k, Cr_{y,n,m}^k \leq 0.3 \quad (35)$$

The time dependency in the Peclet numbers, may be removed by substituting Equation 31a into Equation 34a with $R^{\alpha_1} = 0.0 \text{ ft}^2/\text{sec}$. Thus obtain for $Pe_{x,n,m}^k$ (results for $Pe_{y,n,m}^k$ are analogous)

$$Pe_{x,n,m}^k = \frac{|u_{n,m+1/2}^k| (\mu_1)_m \Delta\alpha_1}{D\sqrt{g} |u_{n,m+1/2}^k| h_{n,m}^k / C_{n,m}^k} = \frac{C_{n,m}^k (\mu_1)_m \Delta\alpha_1}{D\sqrt{g} h_{n,m}^k} \quad (36)$$

Since $C_{n,m}^k$ and $h_{n,m}^k$ vary extremely slowly with time, we may drop the k superscript and obtain

$$Pe_{x,n,m} = \frac{C_{n,m} (\mu_1)_m \Delta\alpha_1}{D\sqrt{g} h_{n,m}} \quad (37a)$$

$$Pe_{y,n,m} = \frac{C_{n,m} (\mu_2)_n \Delta\alpha_2}{D\sqrt{g} h_{n,m}} \quad (37b)$$

In all simulations, the following relations hold for the cell Peclet numbers in the vicinity of the sharp front:

$$Pe_{x,n,m}, Pe_{y,n,m} \geq 100 \quad (38)$$

Results at the end of the simulation for the FTUS scheme are shown in Table I. All concentrations are positive and the initial mass of 0.132435×10^{14} equaled the final mass plus the diffusion of material through the boundaries to within the precision limits of the CRAY I-S.

Results at the end of the simulation for the FTCS scheme are shown in Table II. Since the cell Peclet number limit of 2 is violated in the vicinity of the front, oscillations develop behind the movement of the front. Mass is conserved in the simulation, at the expense of negative concentrations and cell concentrations greater than 10.0.

The three FCT limiters outlined previously were tested. The original Zalesak limiter results are shown in Table III. Cell concentrations greater than 10.0 were developed. The alternative one (mixed time level) limiter results are shown in Table IV. No cell concentrations exceed 10.0. However, Q^+ and Q^- may now be negative unlike in the original limiter. Some small negative concentrations are also developed. The second alternative (previous time level) limiter results are shown in Table V. No

Table I. FTUS Results at $5\tau (\times 10^3)$

M/N	100	101	102	103	104	105	106	107
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	20.0	20.0	20.0	20.0	21.0	0.0	0.0
54	7.0	9998.0	9998.0	9997.0	9997.0	9988.0	0.0	0.0
55	12.0	9961.0	9960.0	9959.0	9959.0	9944.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table II. FTCS Results at $5\tau (\times 10^3)$

M/N	100	101	102	103	104	105	106	107
50	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0
51	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0
53	0.0	10.0	10.0	10.0	10.0	10.0	-0.0	0.0
54	4.0	10009.0	10005.0	10005.0	10005.0	10001.0	-4.0	0.0
55	6.0	9985.0	9979.0	9978.0	9978.0	9970.0	-6.0	0.0
56	-0.	-14.0	-14.0	-15.0	-15.0	-16.0	0.0	-0.0
57	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0

Table III. Original Zalesak Limiter FCT Results at $5\tau (\times 10^3)$

M/N	100	101	102	103	104	105	106	107
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	17.0	16.0	16.0	16.0	10.0	0.0	0.0
54	6.0	10001.0	10001.0	10001.0	10001.0	9996.0	0.0	0.0
55	6.0	9967.0	9960.0	9959.0	9959.0	9944.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.0
57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table IV. Mixed Time Level Limiter FCT Results at $5\tau (\times 10^3)$

M/N	100	101	102	103	104	105	106	107
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0
53	0.0	18.0	17.0	17.0	17.0	11.0	0.0	0.0
54	7.0	10000.0	10000.0	10000.0	10000.0	9995.0	0.0	0.0
55	6.0	9967.0	9960.0	9959.0	9959.0	9944.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57	0.0	0.0	0.0	0.0	0.0	0.0	-0.0	0.0

Table V. Previous Time Level Limiter FCT Results at $5\tau (\times 10^3)$

M/N	100	101	102	103	104	105	106	107
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	20.0	20.0	20.0	20.0	17.0	0.0	0.0
54	7.0	9998.0	9998.0	9998.0	9997.0	9990.0	0.0	0.0
55	7.0	9960.0	9960.0	9959.0	9959.0	9944.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

cell concentrations exceed 10.0, Q^+ and Q^- are nonnegative, and no negative concentrations are developed.

The FCT method results exhibit nonoscillatory profiles with less frontal smearing than the results obtained by the FTUS method even after only five time steps. FTCS methods are unacceptable due to their oscillatory behavior. To obtain oscillation free results with the FTCS method, the space step would be so small that an application to Mississippi Sound would be economically infeasible.

10. CONCLUSIONS AND DIRECTIONS OF FUTURE WORK. For the sharp front problem, the FCT method employing the previous time level limiter was found to be superior to any of the other methods. Based upon this finding, the FCT method employing the previous time level limiter is recommended for further study. Presently work is under way to consider verifying this FCT method and calibrating the dispersion coefficients. A 5- to 6-day time period is being contemplated for the calibration period. Based upon the 6-minute time step used in the sharp front problem, simulations on the order 1200-1440 time steps will be performed.

11. ACKNOWLEDGEMENTS. The work performed here is part of a numerical investigation of Mississippi Sound sponsored by the U. S. Army Engineer, Mobile District. Permission was granted by the Chief of Engineers to publish this information.

12. REFERENCES.

1. Wanstrath, J. J., Nearshore Numerical Storm Surge and Tidal Simulation, Technical Report H-77-17, September 1977, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Mississippi.
2. Butler, H. L., "Evolution of a Numerical Model for Simulating Long-Period Wave Behavior in Ocean-Estuarine Systems," in Estuarine and Wetland Processes, Hamilton, P. and Macdonald, K. (eds.), Plenum Press, New York, 1980.
3. Butler, H. L. et al., Waterways Implicit Flooding Model Documentation, Preliminary Draft, Fall 1981, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Mississippi.
4. Boris, J. P. and Book, D. L., 1973 (Jan), "Flux-Corrected Transport I: SHASTA, A Fluid Transport Algorithm that Works," Journal of Computational Physics, Vol. II, No. 1, pp 38-69.
5. Zalesak, S. T., 1979, "Fully Multi-Dimensional Flux-Corrected Transport Algorithms for Fluids," Journal of Computational Physics, 31, pp 335-362.
6. Leendertse, J. J., A Water Quality Simulation Model for Well Mixed Estuaries and Coastal Seas: Vol. I, Principles of Computation, The Rand Corporation, RM-6230-RC, February 1970.

Grid Generation Techniques for Projectile Configurations

Charles J. Nietubicz
Karen R. Heavey

Launch and Flight Division
U.S. Army Ballistic Research Laboratory
U.S. Army Armament Research and Development Command
Aberdeen Proving Ground, Maryland 21005

Joseph L. Steger

Department of Aeronautics and Astronautics
Stanford University
Palo, Alto, California 94305

ABSTRACT. The determination of accurate projectile aerodynamics is a major area of concern for shell designers involved with new shapes and Ballisticians concerned with developing artillery aiming data. To achieve the desired goals a research effort has been on going within the Aerodynamics Research Branch/BRL to establish a predictive capability for determining projectile aerodynamics. Modern finite difference codes have been applied to the projectile problem and encouraging results have been obtained in transonic^{1,2} and supersonic³ flow. The generation of good computational grids has been a prerequisite for achieving these flow field solutions.

This paper describes a versatile grid generation program which has been developed for standard, hollow and non-axisymmetric projectile shapes. The grid generator makes use of both elliptic and hyperbolic type partial differential equation solvers. The code allows arbitrary grid point clustering along the body surface in areas of anticipated flow field gradients. The outer boundary can also be arbitrarily defined with its own clustering distribution. The grid is then generated between these two boundarys with either straight rays or by use of an elliptic solver. For those cases when the outer boundary is not restricted, the grid can be generated using a hyperbolic solver which adds the additional benefit of an orthogonal mesh.

The mathematical development of the clustering functions and partial differential equation solvers are described and a series of grids are presented which show the versatility of the grid generation program. Grids for ogive-cylinder-boattail configurations, hollow ring airfoil projectiles and non-axisymmetric projectiles are discussed.

1. INTRODUCTION. The numerical solution of the Navier-Stokes^{4,5,6} equations has been successfully applied to a wide variety of problems. The versatility of these methods is in part attributed to the solution of the transformed set of differential equations. Using transformed equations the physical space can be mapped onto a regularly spaced rectangular region for two dimensional flow. This mapping allows for a wide variety of projectile

configurations to be solved using the same basic numerical technique. An example of some characteristic projectile shapes are shown in Figure 1. A standard projectile shape which consists of an ogive cylinder boattail is shown in 1a; a more non-conventional shape but one of considerable interest, the triangular boattail configuration in 1b; and a tubular projectile configuration which has been type classified and is currently in the Army inventory, in 1c. To calculate the flow field for any one of these shapes the first requirement is to develop a suitable finite difference grid for use with the equation solver. The grid generator described in this paper addresses this problem.

Grid generation routines are employed to generate a network of constant ξ and η lines in the physical x - y plane as indicated in Figure 2a. Corresponding uniform values of ξ and η in the computational space define a one to one mapping between points j,k in the physical plane to points j,k in the computational plane as shown in Figure 2b. The mapping functions are described, at least numerically, once $\xi_{j,k}$ and $\eta_{j,k}$ are known in the physical plane as a function of $x_{j,k}$ and $y_{j,k}$. The metric quantities ξ_x , ξ_y , η_x , and η_y needed in the transformed flow equations can then be determined numerically (see, for example, References 4-6).

The grid generation program presented here describes earlier work done by the authors⁷ as well as extensions which include a hyperbolic solver and the addition of more general projectile shapes. The grid generator is modular and begins with a determination of the body shape. The inner body clustering routine is then called to distribute points in the vicinity of previously determined flow field gradients. The next option allows for the insertion of stings for wake modeling, a rear cut or forward cut. If the outer boundary is free or unconstrained as is the case for conventional projectiles, the hyperbolic solver, which generates a smoothly varying orthogonal grid, is called. For those cases where the outer boundary is constrained, as is the case for tubular projectile shapes, the outer boundary clustering routine is called. Once the outer boundary is specified the elliptic solver is called. The grids generated up to this point would be planar and sufficient for axi-symmetric calculations. However for three dimensional flow fields a periodic or non-periodic grid is generated by spinning the planar grid about the symmetry axis. A flow chart of the overall grid program is shown in Figure 3.

The following sections of the paper will present some of the details used for the inner boundary clustering the outer boundary description and interior grid generation.

2. INNER BOUNDARY DESCRIPTION. The body shape can be input to the program by cards, file specification or as a set of x,y ordinates. The data is assumed to be non-dimensional with respect to the diameter or cord depending on the projectile configuration. Additionally, the code can generate a parabolic arc or standard class of projectiles such as sharp or blunt, tangent or secant ogive-nose, cylindrical body, boattail, or spherical cap. Once the body shape is determined the values of x along the body axis are distributed by contiguously combining segments of the clustering function

$$x_j = x_0 + a\psi_j + b\psi_j^2 + c\psi_j^3 \quad \begin{matrix} x_0 < x_j < x_f \\ j_0 < j < j_f \end{matrix} \quad (1)$$

where $\psi_j = (j-j_0)/(j_f-j_0)$ and j is an index value such that points j_0 to j_f lie in the interval x_0 to x_f and $x_{j_0} = x_0$ while $x_{j_f} = x_f$. Equation (1) is used to cluster x_j as a function of j . The user determines the shape of the clustering function by specifying the initial and final increments of x , that is

$$\Delta x_0 = x_{j_0+1} - x_{j_0} \quad (2a)$$

$$\nabla x_f = x_{j_f} - x_{j_f-1} \quad (2b)$$

Since x_0 and x_f are also specified, a , b , and c are determined

$$c = \{\nabla x_f + \Delta x_0 - 2h(x_f - x_0)\}/(h - 3h^2 + 2h^3)$$

$$b = \{\Delta x_0 - h(x_f - x_0) - c(h^3 - h)\}/(h^2 - h)$$

$$a = x_f - x_0 - b - c$$

where $h = (j_f - j_0)^{-1}$.

The amount of clustering at each point is determined by the specified values of Δx_0 and ∇x_f . Moreover, because Δx_0 and ∇x_f are specified, the user can smoothly patch functions together to form a general clustering function. One drawback to the clustering function, Eq. (1), is that the function is not guaranteed to be monotone in the interval. This can happen, for example, if Δx_0 is too small and ∇x_f too large.

At this point a sting or forward cut can be added to the previously described body as shown in Figures 4a and 4b. Again the clustering function of Equation (1) is used to distribute points along these new boundaries.

3. GRID GENERATION USING A HYPERBOLIC SOLVER. For most projectile applications the outer boundary is unconstrained and simply needs to be placed far enough away from the projectile body so as not to adversely affect the flow field solution. This situation represents an ideal case for a hyperbolic grid generation scheme.

Once the body points have been redistributed and the sting or cut has been determined, a grid can be generated using a hyperbolic solver similar to that described in Reference 8. Before the actual solver can be implemented however, the distance to the outer boundary must be specified and either constant spacing in η or some type of stretching function is required. The η stretching used here is determined by the following relationship

$$\Delta s_k = \Delta s_0 (1 + \epsilon)^{k-1}, \quad k = 1, k_{\max} - 1 \quad (3)$$

Here Δs_0 is the minimum specified grid spacing desired at the wall or inner boundary. The parameter ϵ is determined by a Newton-Raphson iteration process so that the sum of the above increments matches the known arc length between $\eta = 0$ and $\eta = \eta_{\max}$ for points which have the same value of ξ .

The governing equations for the hyperbolic solver are obtained by requiring: (1) the coordinate lines ξ and η to be orthogonal; and (2) the specification of a cell volume or area for the two dimensional case. The condition of orthogonality requires

$$\Delta \xi \cdot \Delta \eta = 0 \quad (4)$$

The second equation is obtained by specifying a grid cell volume (or area in two dimensions). Since the grid cell volume is finite the transformation Jacobian will be greater than one, i.e.,

$$dx dy = |x_{\xi} y_{\eta} - x_{\eta} y_{\xi}| d\xi d\eta \quad (5)$$

The set of grid generation equations are therefore given in the physical plane by

$$\xi_x \eta_x + \xi_y \eta_y = 0$$

$$\xi_x \eta_y - \xi_y \eta_x = J$$

or in the transformed plane by (6)

$$x_{\xi} x_{\eta} + y_{\xi} y_{\eta} = 0$$

$$x_{\xi} y_{\eta} - x_{\eta} y_{\xi} = 1/J \equiv V$$

Using local linearization for this set of non-linear differential equations, the resulting system is shown to be hyperbolic⁸ and can therefore be marched in the η direction.

The linearized set of differential equations to be solved numerically is written in vector form as

$$A\vec{r}_\xi + B\vec{r}_\eta = \vec{f} \quad (7)$$

where

$$A = \begin{bmatrix} x_\eta^0 & y_\eta^0 \\ y_\eta^0 - x_\eta^0 \end{bmatrix}, \quad B = \begin{bmatrix} x_\xi^0 & y_\xi^0 \\ -y_\xi^0 & x_\xi^0 \end{bmatrix}$$

$$\vec{f} = \begin{bmatrix} 0 \\ V + V^0 \end{bmatrix}, \quad \vec{r} = \begin{bmatrix} x \\ y \end{bmatrix}$$

where x_η^0, y_η^0 , etc., refers to known conditions.

The set of Equations (7) are solved with an implicit finite difference scheme which is first order accurate in the η direction(k) and where central differencing is used in the ξ direction(j). The resulting set of finite difference equations becomes

$$\frac{A(\vec{r}_{j+1,k+1} - \vec{r}_{j-1,k+1})}{2\Delta\xi} + B\frac{(\vec{r}_{j,k+1} - \vec{r}_{j,k})}{\Delta\eta} = \vec{f}_{j,k+1} \quad (8)$$

Rearranging Eq. (8) and setting $\Delta\eta = \Delta\xi = 1$ results in

$$\frac{A}{2} \vec{r}_{j+1,k+1} + B \vec{r}_{j,k+1} - \frac{A}{2} \vec{r}_{j-1,k+1} = \vec{f}_{j,k+1} + B \vec{r}_{j,k} = \vec{d}_{j,k+1} \quad (9)$$

where

$$\vec{d}_{j,k+1} = \begin{bmatrix} (x_\xi^0 x^0 + y_\xi^0 y^0)_{j,k} \\ (-y_\xi^0 x^0 + x_\xi^0 y^0)_{j,k} + V + V^0 \end{bmatrix}$$

Equation (9) is now in a form which can be easily solved by inverting a block tridiagonal matrix with 2×2 blocks. The terms x_ξ^0 and y_ξ^0 are central differenced as

$$\begin{aligned}
 x_{\xi,j,k}^0 &= \frac{x_{j+1,k} - x_{j-1,k}}{2} \\
 y_{\xi,j,k}^0 &= \frac{y_{j+1,k} - y_{j-1,k}}{2}
 \end{aligned}
 \tag{10}$$

The terms x_n^0 and y_n^0 are obtained from Equation (6) evaluated at the old station(o). That is

$$\begin{aligned}
 x_{\xi}^0 x_n^0 + y_{\xi}^0 y_n^0 &= 0 \\
 x_{\xi}^0 y_n^0 - x_n^0 y_{\xi}^0 &= V^0
 \end{aligned}
 \tag{11}$$

Solving for x_n^0 and y_n^0 with x_{ξ}^0 and y_{ξ}^0 given in (10) yields

$$\begin{aligned}
 x_n^0 &= \frac{-y_{\xi}^0 V^0}{(x_{\xi}^0{}^2 + y_{\xi}^0{}^2)} & y_n^0 &= \frac{x_{\xi}^0 V^0}{(x_{\xi}^0{}^2 + y_{\xi}^0{}^2)}
 \end{aligned}
 \tag{12}$$

The cell volume remains to be specified. This specification is important since it has the effect of controlling the grid evolution as the solution is being marched out from the body. The method chosen here is straight forward and uses the stretching function given by Equation (3). Specifying the minimum spacing at the wall Δs_0 and the total number of points, j_{max} , in the n direction an array of arc lengths Δs_k is determined. Since the Δx is known along the j line, the volumes are calculated by

$$V = (\Delta s_k) (x_{j+1,k} - x_{j,k})
 \tag{13}$$

This specification of cell volumes yields smoothly varying grids in the n direction. Grid volume control is obtained by varying the arc length distribution Δs_k and/or surface point distribution. An additional volume specification approach can be found in Reference 8. A grid generated using this technique is shown in Figure 5a and 5b for a standard projectile configuration with sting.

4. OUTER BOUNDARY DEFINITION. For those cases where the outer boundary is constrained or specified a grid point distribution along the outer boundary is required. An example is shown in Figure 6. A part of the grid generation problem then is the formation of an arbitrary outer boundary. Here this boundary is built up by connecting contiguous cubic segments, which in the degenerate case can be straight lines. Figures 7a and 7b illustrate two

typical outer boundary curves. In Figure 7a three cubic segments make up the boundary, $\eta = \eta_{\max}$. Each segment is formed by specifying the values of x, y , and angle θ , at the endpoints, where θ is the angle between the curve and the x axis. In the example, Figure 7a, $\theta_a = 90^\circ$, $\theta_b = \theta_c = 0^\circ$, or 180° and $\theta_d = 90^\circ$.

The data (x, y, θ) at each endpoint determines the shape of the parametric curves

$$\begin{aligned} x &= x_0 + \alpha_1 t + \alpha_2 t^2 \\ y &= y_0 + \beta_1 t + \beta_2 t^2 \end{aligned} \quad 0 \leq t \leq 1 \quad (14)$$

which are equivalent to a cubic

$$y = y_0 + \gamma_1(x-x_0) + \gamma_2(x-x_0)^2 + \gamma_3(x-x_0)^3 \quad (15)$$

The parametric cubic is used because the condition $\frac{dy}{dx} = \infty$ can be specified (segment bc of Figure 7b has this constraint at both endpoints).

The solution for the parameters $\alpha_1, \alpha_2, \beta_1$, and β_2 can be found in Reference 7.

The outer boundary curve is thus made up of contiguous cubic segments starting from the $\xi = 0$ boundary. Points are distributed along this curve either as a uniform distribution of arc length, or as a specified arc length distribution using the previously defined clustering scheme, Eq. (1). Since the true arc length is not specified a priori, precise alignment of points along the outer boundary can be determined only after the cubic segments are specified and the arc length is computed.

5. STRAIGHT RAY AND ELLIPTIC GRID GENERATION. Once the boundary curves have been specified and points are distributed on the $\eta = 0$ and η_{\max} boundaries, two types of grid generation procedures can be used.

In the first case, lines of constant ξ (i.e., the rays emerging from the body) are formed by simply connecting straight lines from points along $\eta = 0$ to points along $\eta = \eta_{\max}$. The spacing in η along each such line is either uniform or is determined by the stretching relationship given by Equation (3). Figures 8a and 8b illustrate a straight ray grid with clustering in η for a tubular projectile.

In the second case, the grid is generated with elliptic partial differential equations following References 9, 10, and 11. The grid generating equations are solved on the specified computational space for unknowns $x_{j,k}$ and $y_{j,k}$:

$$\begin{aligned}\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} &= -J^2 (\bar{P}x_{\xi} + \bar{Q}x_{\eta}) \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} &= -J^2 (\bar{P}y_{\xi} + \bar{Q}y_{\eta})\end{aligned}\quad (16)$$

where

$$\alpha = x_{\eta}^2 + y_{\eta}^2, \beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}, \gamma = x_{\xi}^2 + y_{\xi}^2, J = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}$$

and

$$\bar{P} = P_0 e^{-a(\eta-\eta_0)} + P_m e^{-a(\eta-\eta_{\max})}$$

$$\bar{Q} = Q_0 e^{-b(\eta-\eta_0)} + Q_m e^{-b(\eta-\eta_{\max})}$$

Here P_0, Q_0, P_m, Q_m, a and b are prescribed clustering parameters. Along the $\eta = 0$ and $\eta = \eta_{\max}$ boundaries, $x_{j,k}$ and $y_{j,k}$ have been previously prescribed. Along the $\xi = 0$ and $\xi = \xi_{\max}$, which are either vertical or horizontal lines in the physical space, the following boundary conditions are enforced: either

$$x \text{ is given and } y_{\xi} = 0$$

on a vertical boundary, or

(17)

$$x_{\xi} = 0 \text{ and } y \text{ is given}$$

on a horizontal boundary.

The difference equations to Eq. (16) (see Reference 7) are solved with a successive line over relaxation (SLOR) procedure. As an initial guess for the relaxation procedure the straight line ray procedure previously described is used. For the most part, if coefficients \bar{P} and \bar{Q} are large, the SLOR procedure is very difficult to converge. Consequently, the algebraic clustering function, Eq. (3) is recommended.

In the algebraic clustering approach the elliptic solver is used to generate a grid with $\bar{P} = \bar{Q} = 0$. The x, y points along a $\xi = \text{constant}$ line are then redistributed along this line as a function of arc length. The clustering function Eq. (3) is used for this purpose. This procedure works quite well and provides excellent control of the grid spacing near the body surface. Further details are given in Reference 7.

The elliptic solver need not be used over the entire range in ξ . Because of the boundary condition, Eq. (17), the elliptic equations can be joined to a straight ray along any vertical or horizontal boundary line in ξ . Figure 9 shows details of such a procedure used for a secant-ogive-cylinder boattail projectile which also includes a sting. Here the ξ -region over the secant-ogive nose is generated using the elliptic equations while the remainder is meshed with straight rays. After the basic grid is formed, the entire grid is clustered in n using Eq. (3).

6. 3D GRIDS. The final option available in the code is the ability to generate three dimensional grids. At present the grids are formed in a two dimensional plane and then rotated about a symmetry axis. The rotation is either periodic or non-periodic depending on the grid desired. For cases where the flow field has planar symmetry, such as a projectile at angle of attack, without spin, a non-periodic grid is generated.

The generation of grids for projectile shapes, with non-axisymmetric sections (Figure 1b) is accomplished with a series of planar grids. Planes are generated normal to the projectile axis at incremental values of Δx . For each of these planes a grid is generated using an O type grid (Figure 10). These grids are then combined to form a three dimensional mesh making sure that continuity in the x direction is maintained.

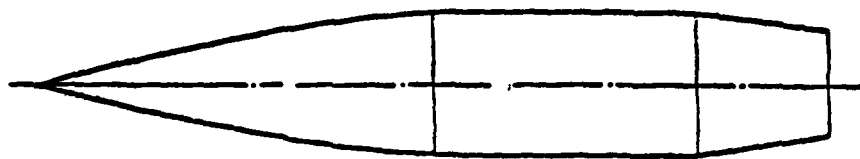
7. SUMMARY. A versatile grid generation program has been described which utilizes general elliptic and hyperbolic equation solvers for internal grid generation. The flexibility of longitudinal grid point distribution is obtained with the general clustering functions allowing points to be placed in the vicinity of flow field gradients. Grid clustering is also obtained near the body surface for viscous flow field calculations.

A series of grids have been presented which show the versatility of the code. Grids for secant-ogive-cylinder boattails have been shown using an elliptic solver, hyperbolic solver and a hybrid elliptic/straight ray solver. The generation of a grid for a non-conventional hollow projectile shape has been demonstrated.

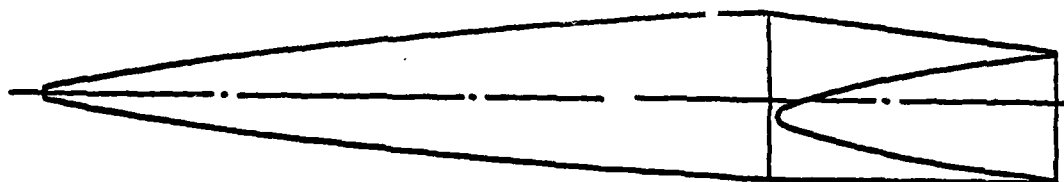
8. ACKNOWLEDGEMENTS. The authors would like to thank Dr. Harry Dwyer for his helpful discussions concerning the hyperbolic solver.

References

1. Nietubicz, C. J., "Navier-Stokes Computations for Conventional and Hollow Projectile Shapes at Transonic Velocities", AIAA Paper No. 81-1262, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA, 1981.
2. Nietubicz, C. J., Inger, G. R., and Danberg, J. E., "A Theoretical and Experimental Investigation of a Transonic Projectile Flow Field", AIAA Paper No. 82-0101, AIAA 20th Aerospace Sciences Meeting, Orlando, FL, January 1982.
3. Schiff, L. B., and Sturek, W. B., "Numerical Simulation of Steady Supersonic Flow Over Cone Ogive-Cylinder-Boattail Body", AIAA Paper No. 80-0066, January 14-16, 1980.
4. Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries", AIAA Journal, Vol. 16, July 1978, pp. 679-686.
5. Pulliam, T. H., and Steger, J. L., "On Implicit Finite-Difference Simulations of Three-Dimensional Flow", AIAA Paper No. 78-10, 1978.
6. Nietubicz, C. J., Pulliam, T. H., and Steger, J. L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", AIAA Paper No. 79-0010, 1979.
7. Steger, J. L., Nietubicz, C. J., and Heavey, K. R., "A General Curvilinear Grid Generation Program for Projectile Configurations", BRL Memorandum Report-MR-03142, October 1981.
8. Steger, J. L., and Chaussee, D. S., "Generation of Body Fitted Coordinates Using Hyperbolic Differential Equations". Flow Simulations Report 80-1, January 1980.
9. Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain". Journal of Comp. Physics, Vol. 8, 1971, pp. 392-408.
10. Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies". Journal of Comp. Physics, Vol. 15, 1974, pp. 299-319.
11. Sorenson, R. L., and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations". NASA TM 73252, August 1977.

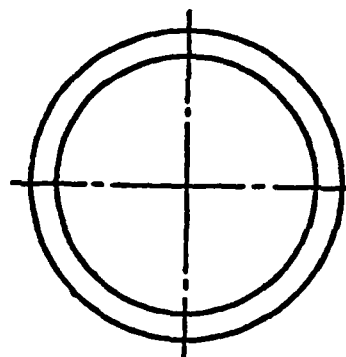
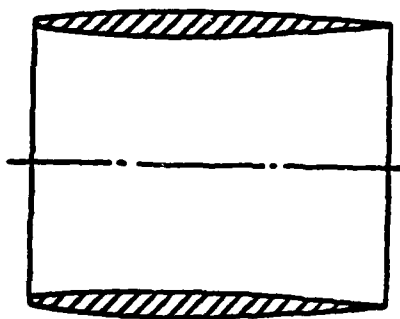


a. Conventional Secant-Ogive Cylinder Boattail (SOCBT) Projectile



7° TRIANGULAR BOATTAIL

b. Secant-Ogive Triangular Boattail Projectile



c. Tubular Projectile

Figure 1. Projectile Configurations

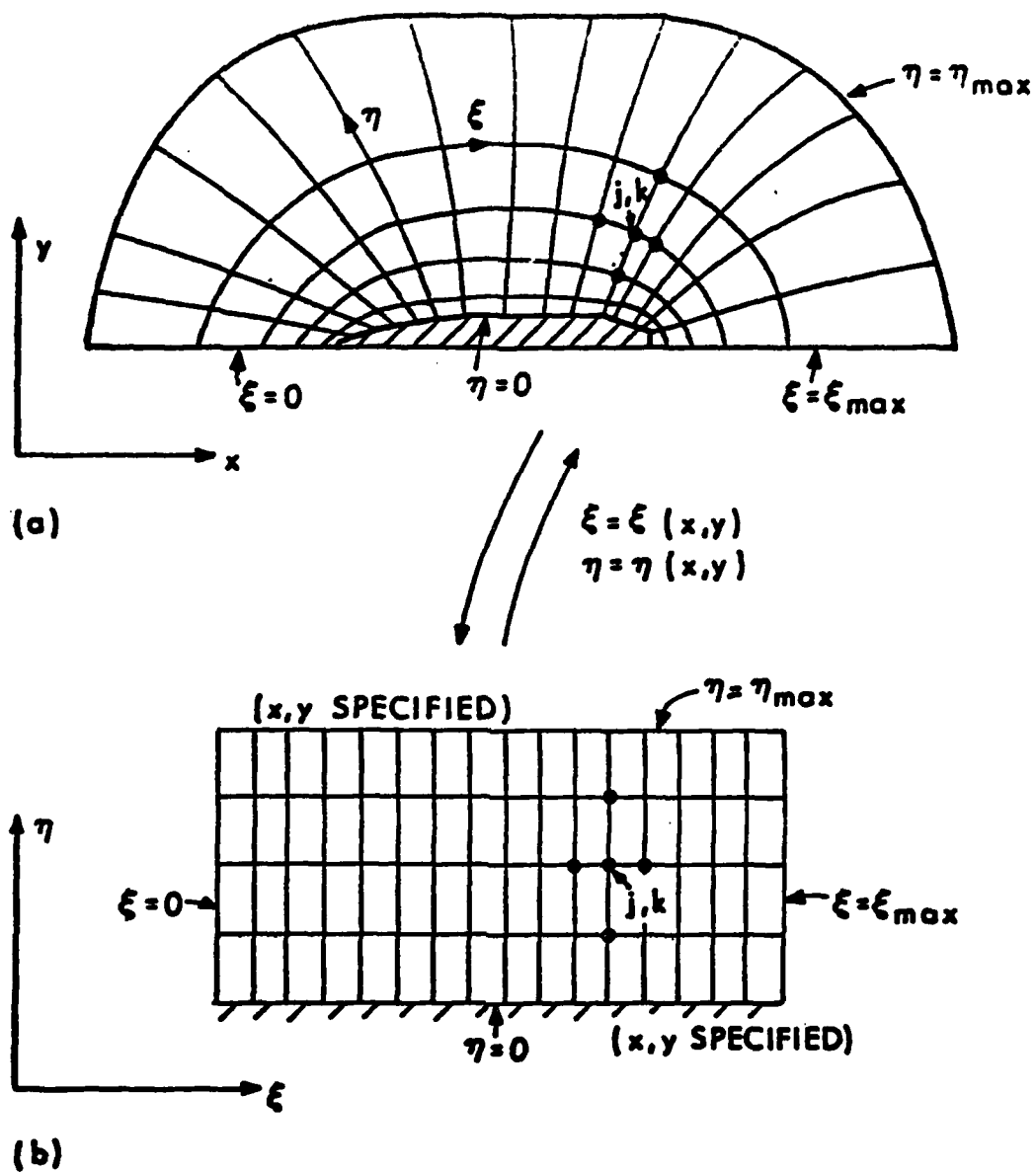


Figure 2. Mapping from Physical Space to Computational Space

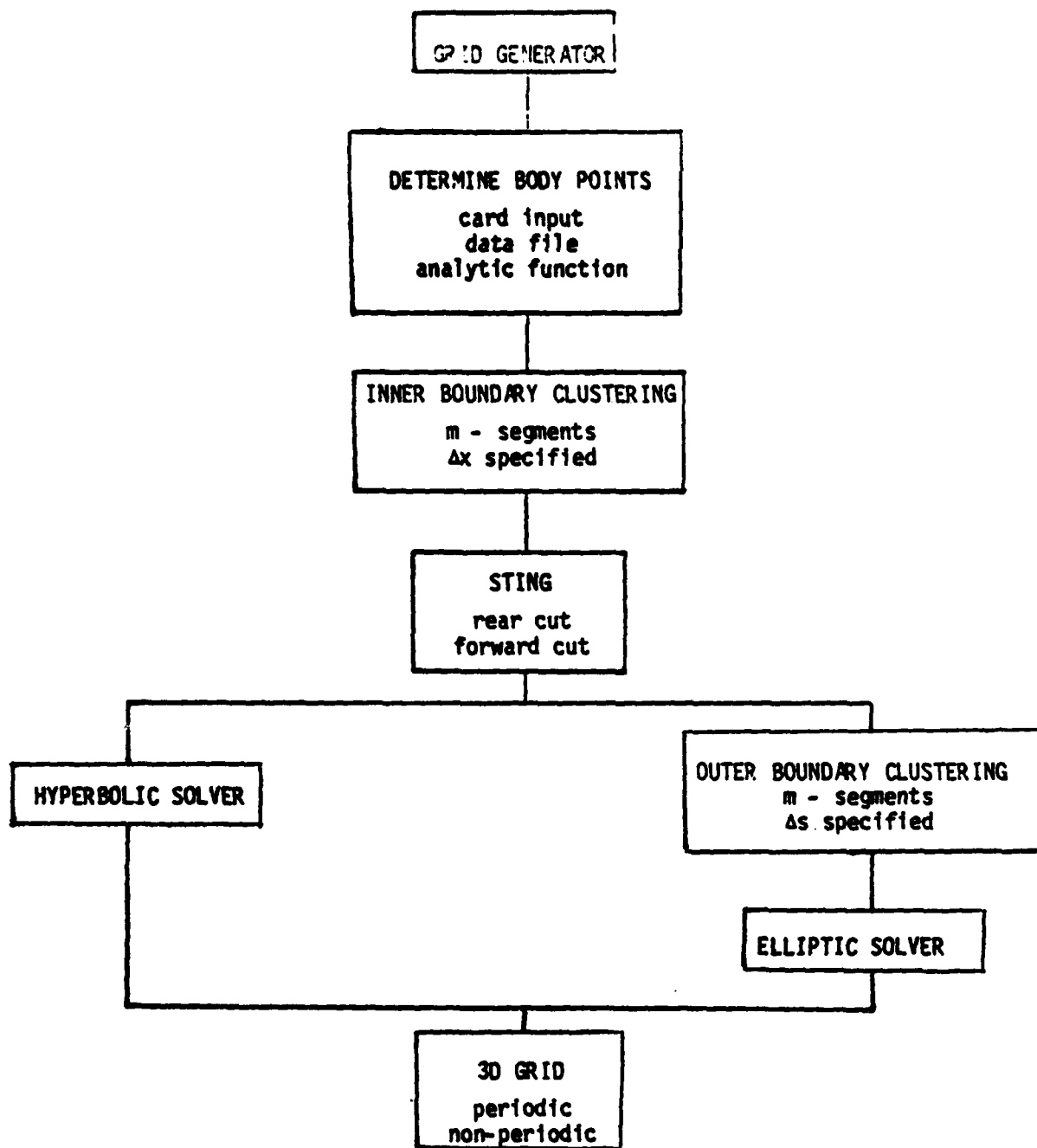


Figure 3. Flow Chart of Grid Generator Programs

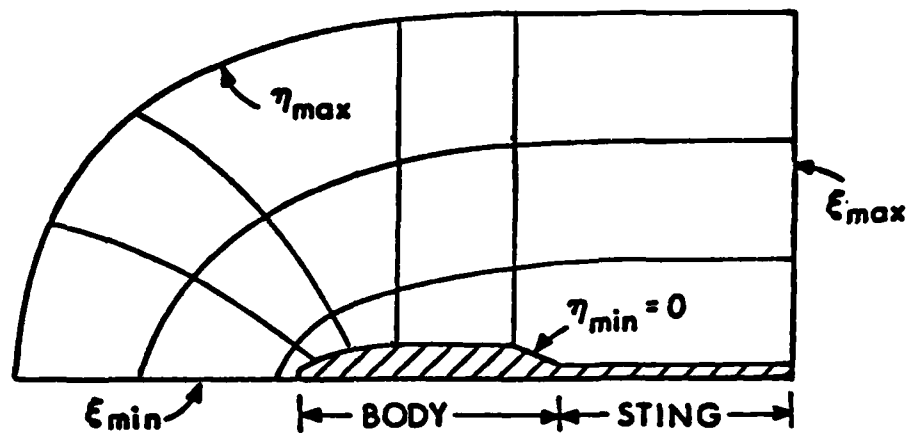


Figure 4a. Standard Projectile Grid with Sting

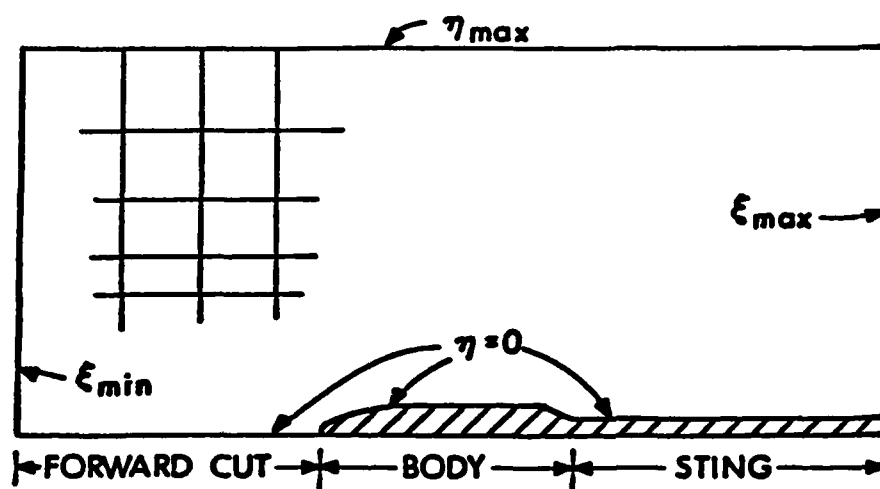


Figure 4b. Cartesian-like Projectile Grid with Sting

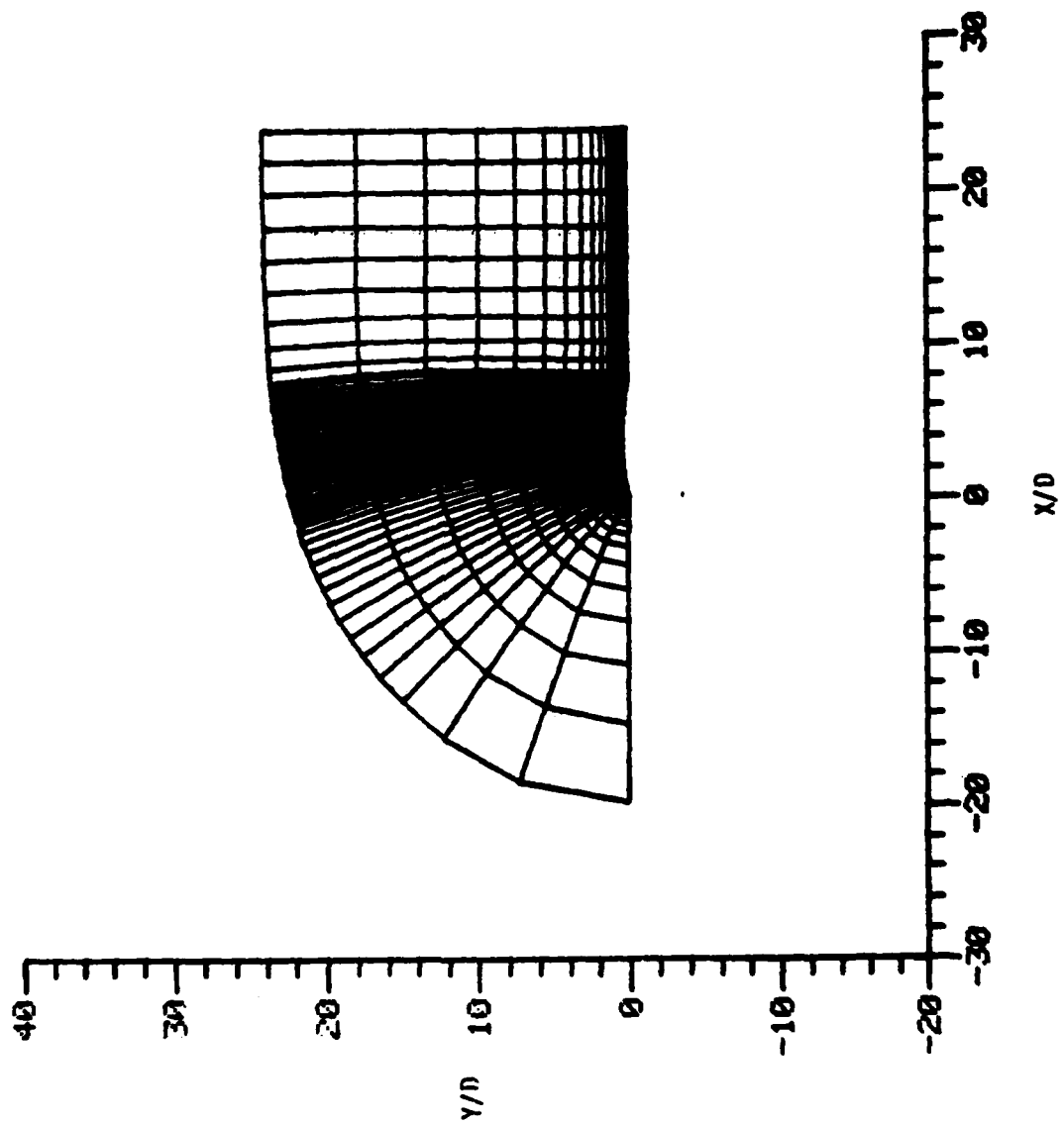


Figure 5a. Overview of Hyperbolic Grid for SOCBT With Sting

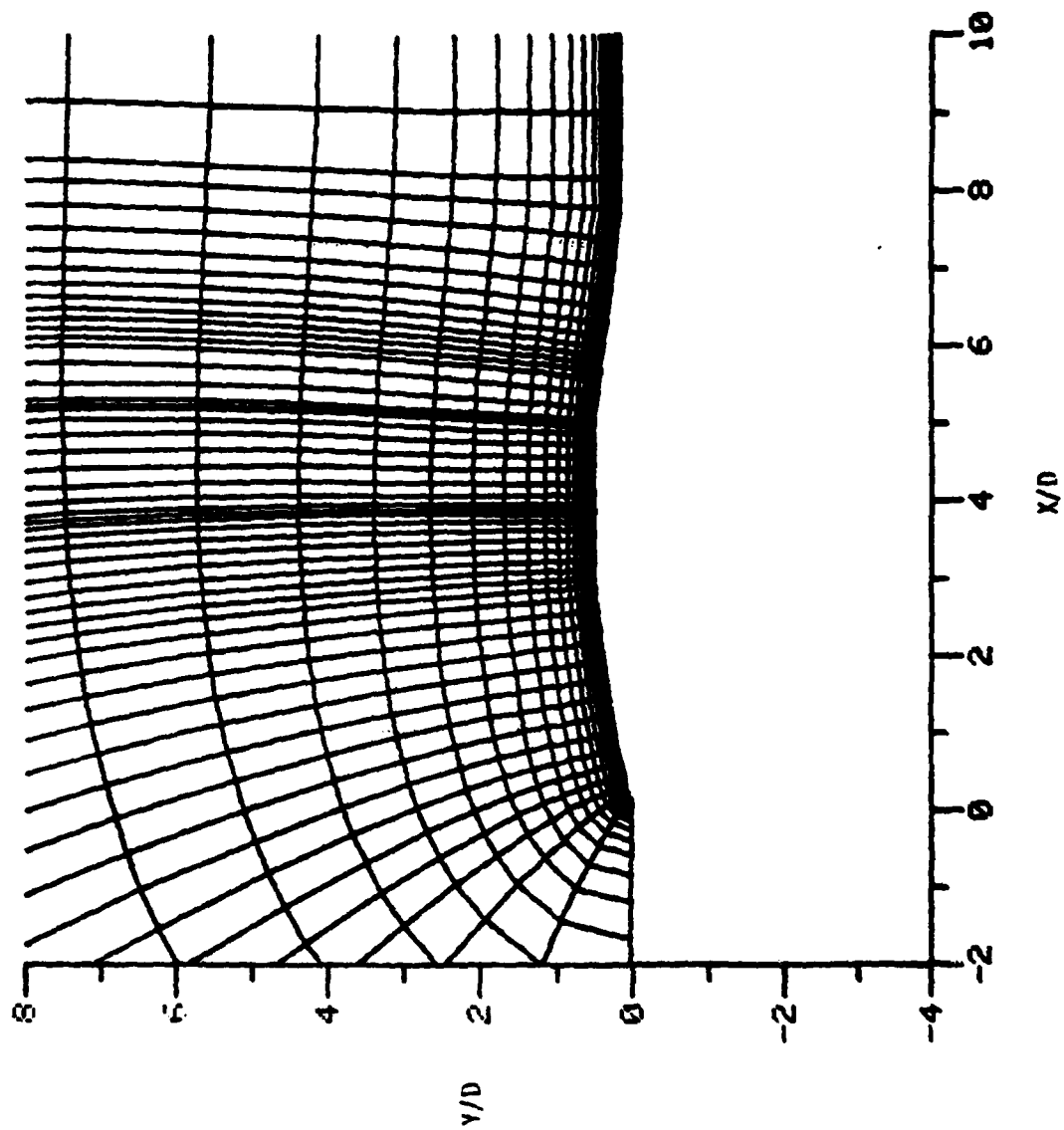


Figure 5b. Expanded View

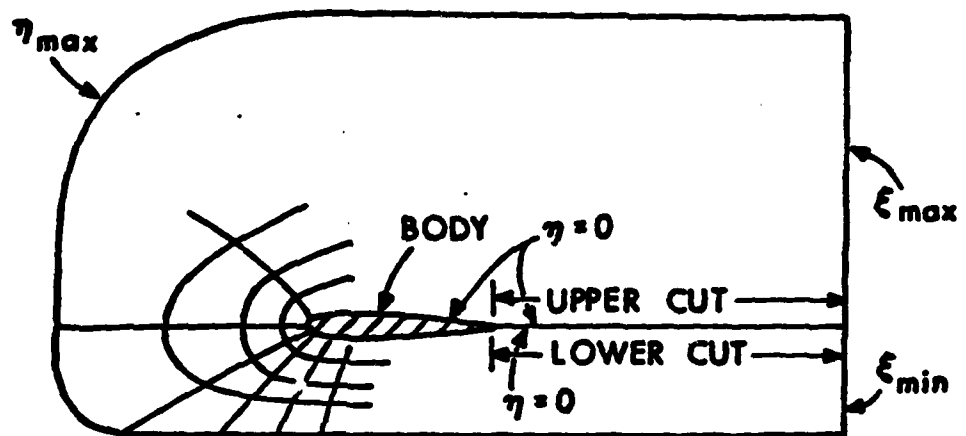
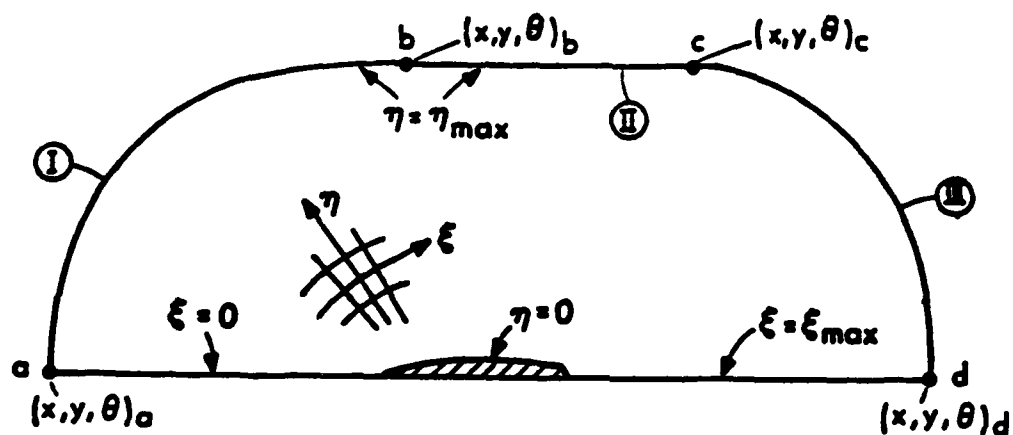
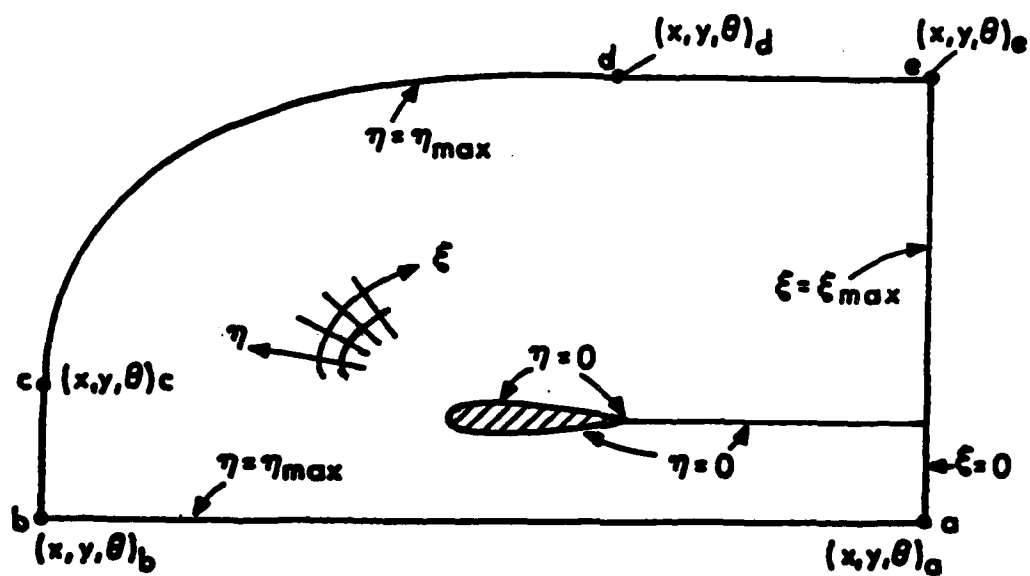


Figure 6. Tubular Projectile Grid or C-Grid



(a) STANDARD PROJECTILE GRID



(b) C-GRID FOR TUBULAR PROJECTILE

Figure 7. Outer Boundary Structure and Terminology for Two Classes of Grid

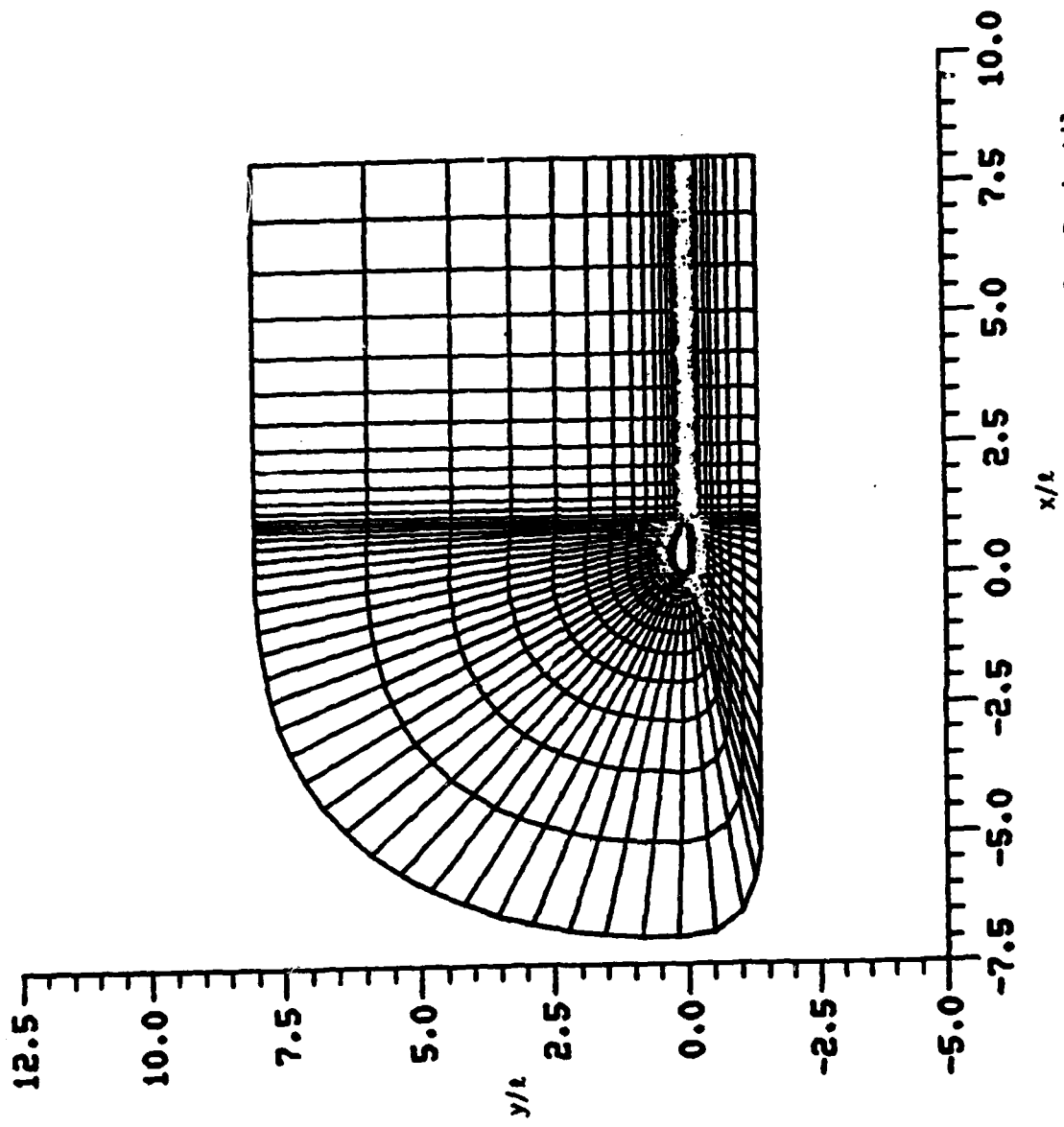


Figure 8a. Overview of Straight Ray Grid for Tubular Projectile

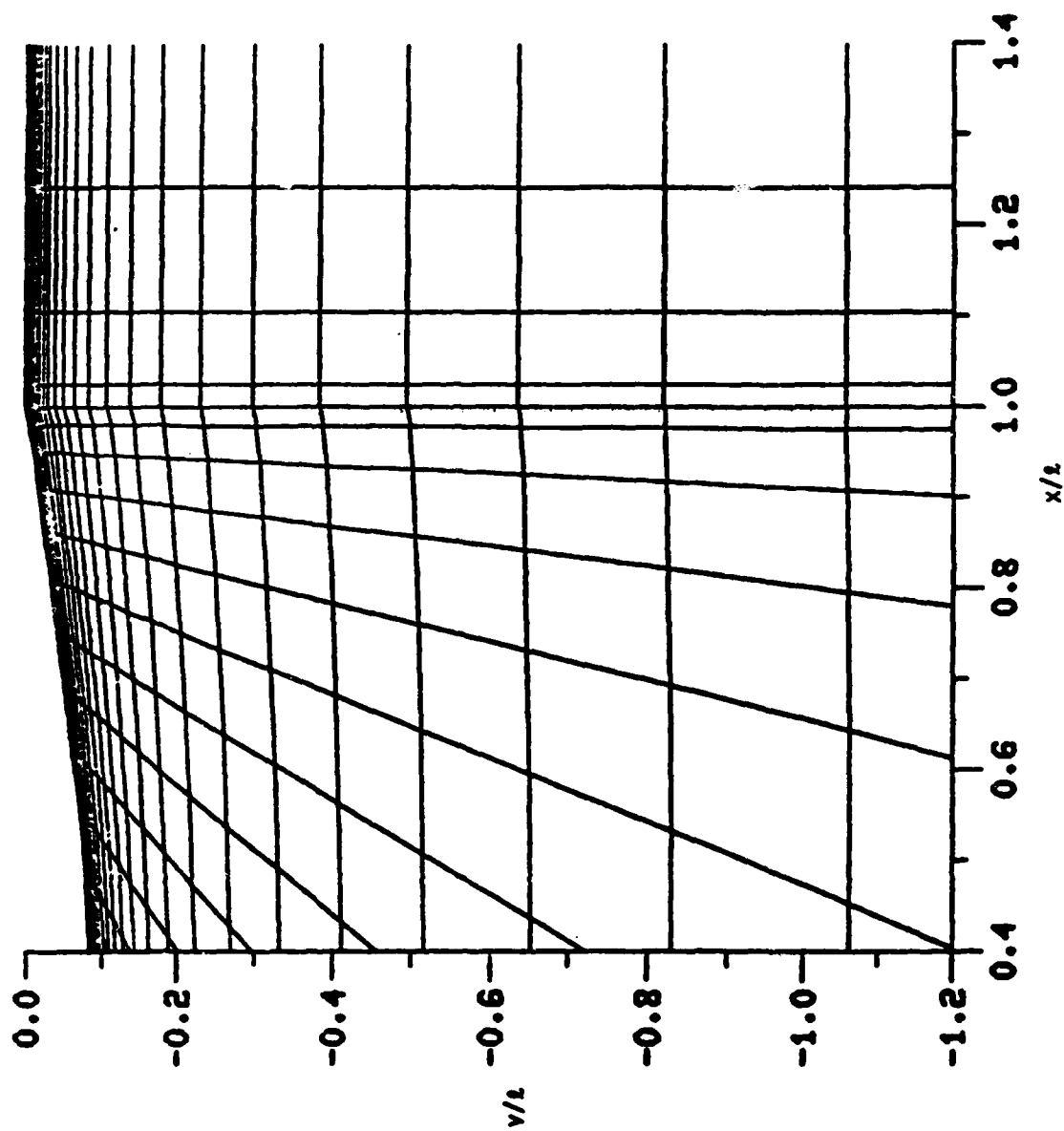


Figure 8b. Grid Detail near Lower Trailing Edge

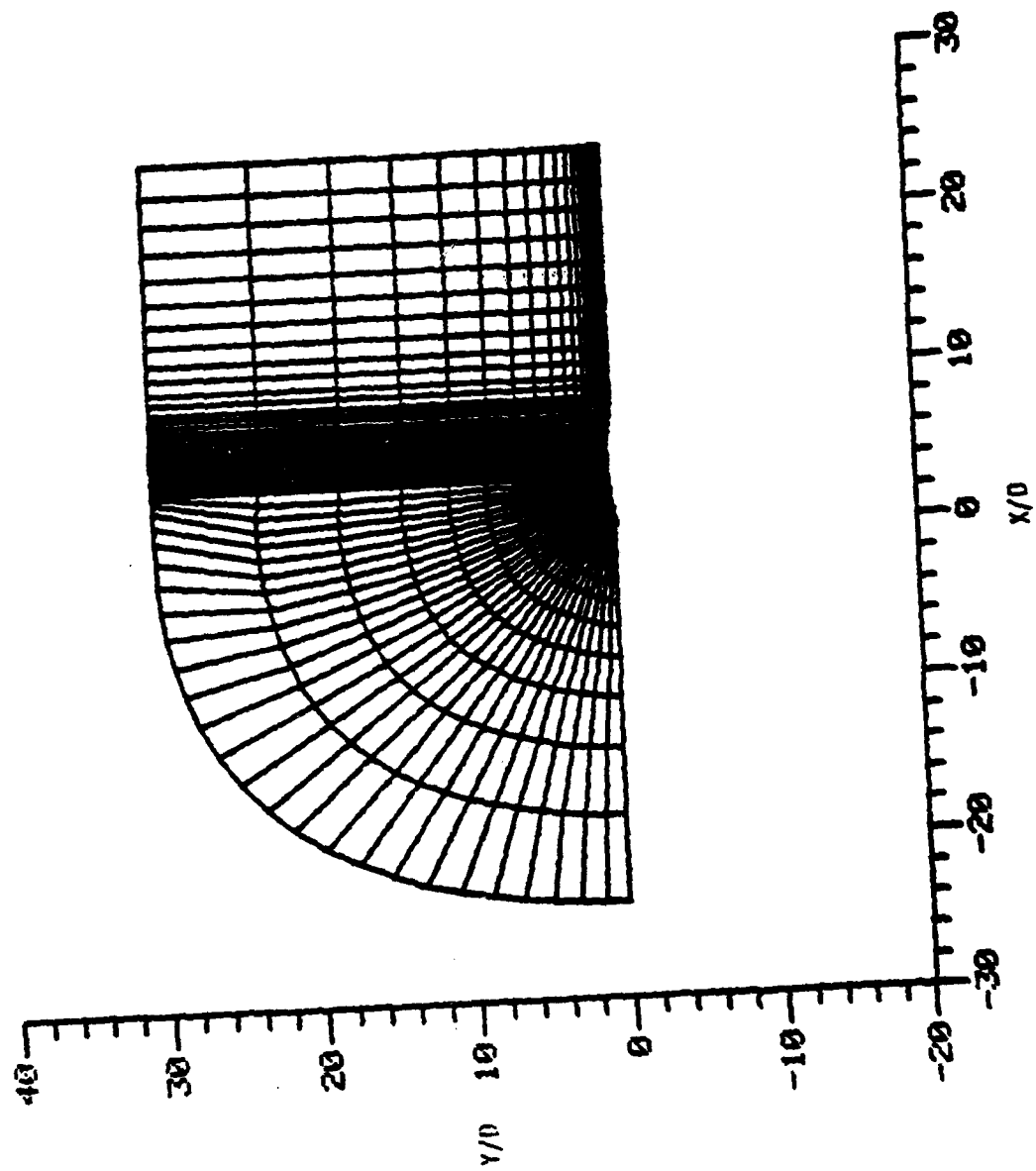


Figure 9a. Hybrid Elliptic and Straight Ray Grid for SOCBT With Sting

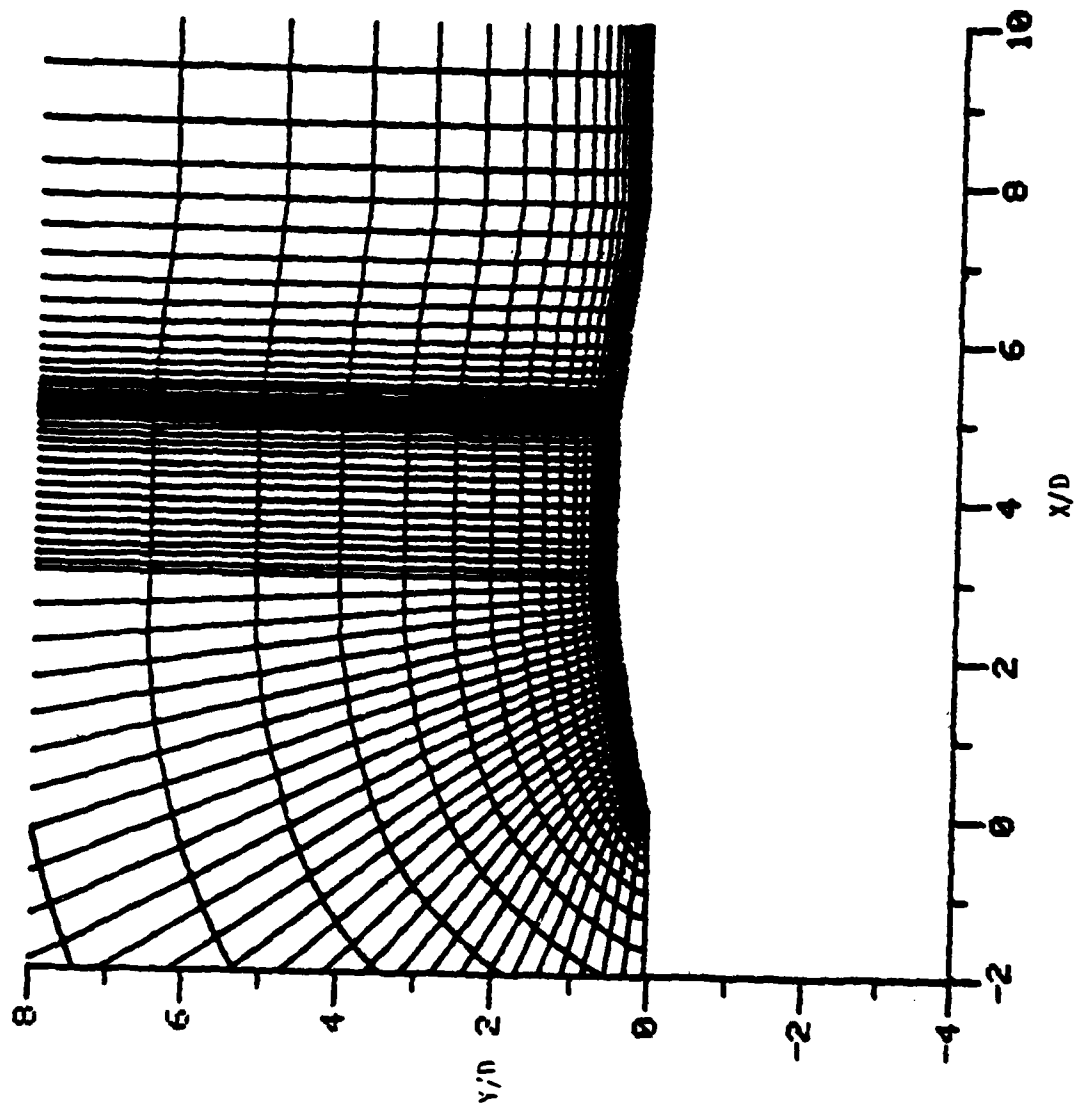


Figure 9b. Expanded View

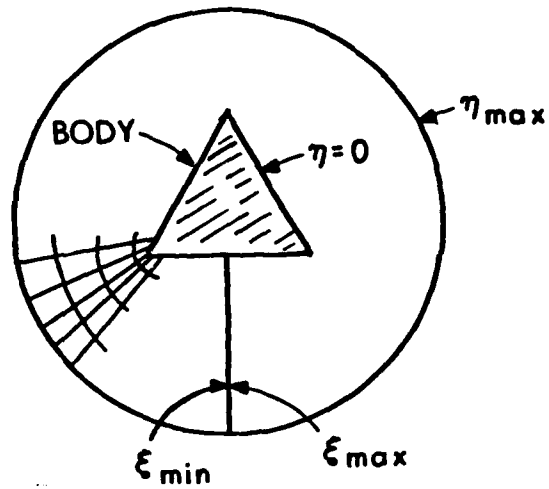


Figure 10a. Projectile Cross Section with Periodic B.C. (O-Grid)

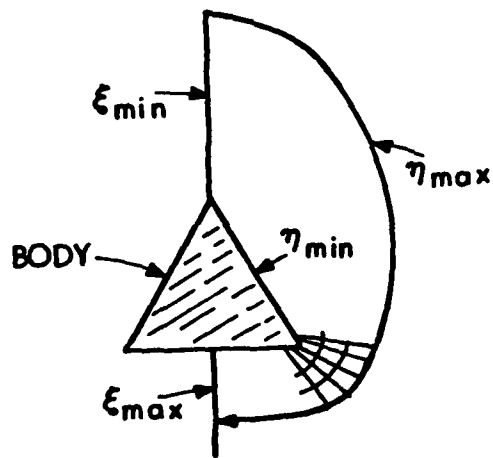


Figure 10b. Projectile Cross Section with Symmetry Plane (O-Grid)

APPLICATION OF RELATIVE COORDINATES IN HYDRODYNAMICS

R. H. Multer
Waterways Experiment Station
Vicksburg, MS

ABSTRACT. A method for approximating the solution of the exact hydrodynamic wave problem is presented.

1. Introduction. It is assumed that the fluid in question is inviscid, of constant density and incompressible and that the flow is irrotational. The case of two-dimensional flow, restricted to a vertical plane, is considered.

In modern times (see Refs 1 and 2) perturbation expansions have been used to develop approximate mathematical models of the hydrodynamic wave problem, i.e., free-surface flows of inviscid, constant-density, incompressible fluids. In most instances these models cannot be solved exactly and it is essential to resort to numerical computation to obtain an estimate of the wave motion. Involved then are three layers of approximation: those about the fluid, those in the perturbation expansion, and those in the numerical approximation. Thence, when the agreement between physical observation and predicted wave behavior are not overly good it is infeasible to identify the culprit.

Presented here is a method of numerically approximating the solution of the exact hydrodynamic wave problem. The most obvious difference between the two is that the intermediate approximation of the perturbation expansion is avoided. Thus comparison between physical observations and computed wave behavior becomes, essentially, an assessment of the validity of the physical assumptions about the process. Madsen and Mai's development (Ref 2) is at least superficially similar to that presented here, i.e., a velocity potential is introduced in both cases and a polynomial is introduced for the velocity

potential in both cases. Madsen and Mai treat this series as a power (Taylor) series.

$$\phi(x,y,t) = \sum_{n=0} \phi_n(x,t) (y+b(x))^n$$

In order for this series to be a harmonic power-series the elevation of the channel bottom, $b(x)$, must be an analytic function (see Ref 2). In most instances this will not be the case. Hence their approximation cannot converge to the solution of the problem. We should note also that power series (see Ref 3 and Ref 4) have relatively slow and weak convergence properties. We shall apply Ritz's method to determine the value of the functions $\{\phi_n\}$. The associated convergence is more rapid (see Ref 5 and Ref 6).

2. Development of the Hydrodynamic Wave Problem. There are two coordinate systems of importance in classical dynamics. The Eulerian coordinate system which describes the behavior of a substance as points fixed in inertial space and the Lagrangian coordinate system. In the dynamics of systems of discrete particles we have the location of the i th particle.

$$x_i = f_i(t)$$

its velocity

$$v_i = d_t x_i = f_i'(t)$$

and its acceleration

$$a_i = d_t v_i = d_{tt} x_i = f_i''(t)$$

The Lagrangian coordinate system is the extension of these definitions for a system of discrete particles to a continuum. Specifically

$$\text{Displacement} \quad x = x(\alpha, \beta, t) \quad (1)$$

$$\text{Velocity} \quad v = \partial_t x(\alpha, \beta, t) \quad (2)$$

$$\text{Acceleration} \quad a = \partial_{tt} x(\alpha, \beta, t) \quad (3)$$

The Lagrangian and Eulerian Coordinate Systems are related by this chain rule.

$$\partial_t \phi(\alpha, \beta) = \partial_t \phi(x, y) + \partial_x \phi \partial_t x + \partial_y \phi \partial_t y$$

which because of Eq 2 may be rewritten as

$$D_t \phi = \partial_t \phi + u \partial_x \phi + v \partial_y \phi \quad (4)$$

where the notation $D_t \phi$ is introduced for clarity.

The Dynamic Equation of Bernoulli, written in Eulerian coordinates, is

$$\partial_t \phi + \frac{1}{2}(\phi_x^2 + \phi_y^2) + \frac{p}{\rho} + gy = C \quad (5)$$

where ϕ is the velocity potential and

$$\phi_x = \partial_x \phi = u, \quad \phi_y = \partial_y \phi = v \quad (6)$$

Eq 6 implies that

$$\frac{1}{2}(\phi_x^2 + \phi_y^2) = u \partial_x \phi + v \partial_y \phi - \frac{1}{2}(\phi_x^2 + \phi_y^2) \quad (7)$$

In line 8 the notation $\phi_{(\alpha, \beta)}$ means that α, β are to be held fixed.

Thence on substituting from Eq 7 into Eq 5 and recalling Eq 4 we have

$$D_t \phi + \frac{P}{\rho} + gy - \frac{1}{2} (\phi_x^2 + \phi_y^2) = C \quad (8)$$

Equation 8 has been used previously by the author (Ref 7) in the study of wave motion in a channel of constant depth.

The difference between Eq 5 and Eq 8 should be kept clearly in mind. Eq 5 gives the rate of change of ϕ at a fixed point inertial space where Eq 8 gives the rate of change of ϕ following a particle. For wave motions in elastic bodies the displacement of particles could be expected to be typically quite small and $\partial_t \phi$ to be a good approximation of $D_t \phi$. For large-amplitude, water-waves the displacements would be large and this would not be the case. This is mentioned because there is an occasional paper on water-waves where the distinction is not recognized.

Considered next are mixed coordinate systems. Suppose that $\partial_x y(\beta, t)$ is finite so that

$$y = y(x, \beta, t) \quad (9)$$

and

$$\psi(x, y, t) = \psi(x, \beta, t) \quad (10)$$

are well behaved functions. Applying the chain rule to Eq 9

$$\partial_t y(\alpha, \beta) = \partial_t y(x, \beta) + \partial_x y \partial_t x(\alpha, \beta)$$

or

$$D_t y = v = \partial_t y + u \partial_x y \quad (11)$$

and similarly

$$D_t \Psi = \partial_t \Psi(\beta) + u \partial_x \Psi(\beta) \quad (12)$$

Before proceeding further it seems advisable to consider the continuum hypothesis which is fundamental to classical mechanics. The following statement is due to Stoker (Ref 8).

THE CONTINUUM HYPOTHESIS: The motion of a substance can be described as a topological deformation which depends continuously on time.

The implication of the continuum hypothesis is that particles which are on the free-surface or the bottom of the channel remain there. Thence we may interpret

$$\eta(x,t) = y(\alpha, \beta, t)_{(\rho = \text{const})}$$

as a mathematical parametric description of a Lagrangian surface. Eq 11 then becomes, because of Eq 2

$$D_t \eta = v = \partial_t \eta + u \partial_x \eta \quad (13)$$

or

$$\partial_t \eta = v - u \partial_x \eta \quad (14)$$

Also, on substituting from Eq 12 into Eq 8

$$\partial_t \phi(x, \beta) + gy + \frac{p}{\rho} - \frac{1}{2}(u^2 + v^2) + u \partial_x \phi(\beta) = C \quad (15)$$

A physical interpretation of the coordinate system may be useful.

Suppose that a vertical wire passed downward through the fluid and that a cork was free to move up and down the wire. Equation 14 would describe the location of the cork and Eq 15 would describe the time rate of change of ϕ following the cork. Let us compare Eqs 5, 8, and 15. Equation 5 describes the variation of ϕ at fixed Eulerian points whereas we shall want to describe the variation of ϕ following particles in the free-surface. Equation 8 has two (sometimes unavoidable) unfavorable properties. First, if there is a net flux through the system, particles must be continuously added to and deleted from the system in numerical computations, because only a finite length of channel may be modeled numerically. Secondly, a set of particles which are initially at some fixed horizontal distance apart do not necessarily remain so and this of course leads to an obvious problem in applying numerical techniques. Equation 15 is introduced to overcome these two problems. Equation 15 becomes unattractive when one end of the channel is bounded by a mechanical generator or a sloping wall which pierces the free-surface. These last two circumstances are of a complexity which precludes their treatment here.

3. STATEMENT OF THE PROBLEM AND SOLUTION ALGORITHM

We shall treat the specific problem of sloshing in a two-dimensional basin bounded by vertical walls.

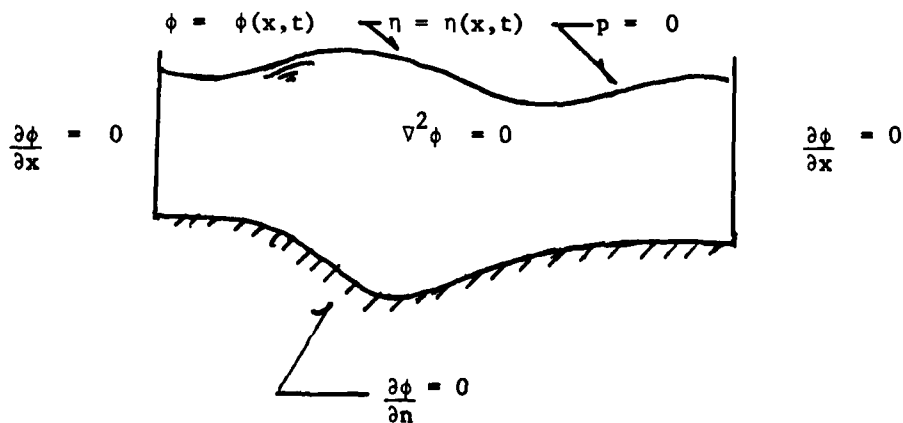


FIG 1

The problem is depicted on Fig 1. It is supposed that at some initial time

$$\eta(x,0) = f_1(x) \quad (16)$$

$$\partial_y \phi(x,\eta,0) = f_2(x) \quad (17)$$

where f_1 and f_2 are known functions. By solving a boundary value problem of the second kind for the Laplace equation

$$\partial_{xx} \phi + \partial_{yy} \phi = 0 \quad (18)$$

and then performing some ancillary computations we may determine ϕ, u , and v at time $t = 0$ along the free surface. Equations 14 and 15 may be written as

$$\partial_t \eta = v - u \partial_x \eta \quad (19)$$

$$\partial_t \phi(x,\eta) = C - g\eta + \frac{1}{2}(u^2 + v^2) - u \partial_x \phi(\eta,t) \quad (20)$$

where every term in the right hand side is known so that we may use those relations to estimate numerically the location of the free surface and the distribution of ϕ along it at time $t = \delta t$. At this point the mixed boundary value problem for the Laplace Equation

$$\partial_{xx} \phi + \partial_{yy} \phi = 0 \quad (21)$$

$$\phi(x,\eta) = \phi(\beta) \quad (\text{on the free-surface}) \quad (22)$$

$$\partial\phi/\partial n = 0 \quad (\text{elsewhere}) \quad (23)$$

where $\phi(\beta)$ is the just computed distribution of ϕ along the free surface, must be solved. Thereinafter we may continue to estimate values of ϕ and η at still later times by cycling through Eqs 19 and 20 and the mixed boundary value problem.

4. Approximate Solution of the Mixed Boundary Value Problem. The remainder of this paper is devoted to the approximate solution of the mixed boundary value problem. It is convenient to introduce the relative coordinate

$$\zeta = \frac{y-b}{h} \quad (24)$$

where the depth h is

$$h = \eta - b \quad (25)$$

thence

$$\zeta_x = -\frac{b+h_x\zeta}{h} \quad \zeta_y = 1/h \quad (26)$$

and, also

$$\begin{aligned} I &= \iint_R (\phi_x^2 + \phi_y^2) dx dy \\ &= \iint_{R^*} \phi_x^2 + 2\zeta_x \partial_\zeta \phi \partial_x \phi + (\zeta_x^2 + \zeta_y^2) \phi_\zeta^2 dx d\zeta \end{aligned} \quad (27)$$

When R is the y -simple region occupied by the fluid and because of Eq 24 R^* is the strip of unit height.

It is assumed that

$$\phi = \phi_{(n)}(x, t) + \sum_n \phi_n(x, t)(1-\zeta)^n \quad (28)$$

This expression satisfies apriori the free-surface boundary condition

$$\phi(x, n, t) = \phi_{(n)}(x, t) \quad (29)$$

and represents a "relatively" complete class of functions. Equation 27 is the functional associated with the Laplace equation and the natural boundary condition $\partial\phi/\partial n$. Finding an extremal of Eq 27 over the relatively complete class of functions given by 28 is, thence, equivalent to finding a solution of the Laplace equation such that $\partial\phi/\partial n$ along the boundary except along the free-surface where ϕ is constrained, i.e., this is the solution of the mixed boundary value problem. Moreover, (see, Ref 6) applying Ritz's method results in a sequence of approximations which converges to the solution of the problem.

Substituting from exps. 28 into Eq 27 and integrating with respect to ζ results in

$$I = \int G + E_m \phi_m + F_m \phi_m + A_{mn} \phi_m \phi_n + B_{mn} \phi_m \phi_n + C_{mn} \phi_m \phi_n dx \quad (30)$$

Hence, because of Euler's Equation

$$\begin{aligned} \frac{d}{dx} (A_{mn} \phi'_n + B_{mn} \phi_n + E_m) \\ - B_{mn}^T \phi'_n - C_{mn} \phi_n - F_m = 0 \end{aligned} \quad (31)$$

provides the extremum of I. Associated with the end boundary conditions are the boundary conditions

$$\phi'_n = 0 \quad (\text{at the ends of the channel}) \quad (32)$$

Equations 31 and 32 then constitute a two-point boundary value problem.

The coefficient matrices of Eq 31 are nonconstant and it is unlikely that a closed form solution of the two-point boundary value problem exists except as the simplest cases. We shall therefore consider the problem of solving Eq 31 numerically. The simplest problem which might be considered would be for the region

$$b(x) = 0, \quad h(x) = \eta(x) = 1 \quad (33)$$

Retaining only one term in the expansion, Eq 31 becomes

$$\frac{1}{3}\phi_1'' - \phi_1 = \frac{1}{2}\phi_1''(\eta) \quad (34)$$

the solution of the homogeneous equation is then

$$\phi_1 = ae^{\sqrt{3}x} + be^{-\sqrt{3}x} \quad (35)$$

This simple result tells us a great deal. Specifically, we see that because of the exponential factors "shooting methods" for solving the two-point boundary value problem would be unstable and therefore finite difference methods are more appropriate. Also we might note that when ϕ'' is replaced by

$$\phi_1'' = \frac{\phi_1^+ + \phi_1^- - 2\phi_1}{2\delta x}$$

the spectral radius of the coefficient matrix in the finite difference formulation is less than 1. Hence this matrix is non-singular and the existence of a solution insured.

5. Local Approximation. The theory presented to this point would seem to be of academic value in ascertaining how accurately hydrodynamic wave theory describes real wave phenomena. It would also appear economically feasible to solve two-dimensional wave problems numerically when real world considerations warranted. The theory presented may immediately be extended to three-dimensions. The problem is, Eq 31 then becomes a system of partial differential equations and the solution of them using contemporary computers would usually be too expensive. Approximation of the solution of Equation 31 is then of some interest.

If one assumes that for all quantities in question

$$\left| \frac{\partial^n \psi}{\partial x^n} \right| < \lambda^n, \quad \lambda \ll 1$$

The following approximation appears reasonable. First approximation

$$C_{mn} \phi_n^{(0)} = F_m - E_m'$$

Higher Order

$$C_{mn} \phi_n^{(k)} = (F_m - E_m' - B_{mn}^T \phi_n' + \dots) \quad (k-1)$$

The exact solution of the first approximation is

$$\phi = \phi_{(\eta)} + \phi_{(\eta)}'' \left(h(\eta - y) - \frac{1}{2}(\eta - y)^2 \right) - b' \phi_{(\eta)}' (\eta - y) + O(\lambda^4)$$

The next approximation has been worked out. The obvious difficulty with it is that fourth order derivatives of the variables become involved.

6. Summary. A method for approximating the solution of the exact hydrodynamic wave problem has been presented. This method avoids the use of a linear additive (Taylor's) series and perturbation approximation. It should, therefore, provide a better approximation to the solution of the problem in question.

Several experiments involving the solution of mixed boundary value problems on the unit strip have been made. These may be compared to linear wave theory. Retaining 4 terms in the series (Eq 28)--actually the odd terms go out in this case--it was found that, for a λ value ($\lambda = 2\pi h/L$) value of 0.3, Ritz's method predicted the correct wave speed to four decimal places, while a corresponding power series was accurate to less than three decimal places. Collocation gave an estimate (using ζ values of 0.25 and 0.75) which was substantially more accurate than the power series estimate but less accurate than Ritz's method. What this computation suggests, beyond the obvious, is that something on the order of 4 or 6 terms need to be retained in Eq 28. For a given number of terms, the degradation of the approximate solution is gradual and increases with h . Hence the shorter the relative wave length, the more terms that need to be retained.

REFERENCES

1. Peregrine, D. H., "Long Waves on a Beach," JFM, Vol 27, 1967.
2. Madsen, O. S., and C. C. Mei, "Dispersive Waves of Finite Amplitude," MIT, Sch of Eng Report No. 117.
3. McShane, J. E., and T. A. Botts, Real Analysis, Van Norstrand, 1959.
(See, Weierstrass Thin and Bernstein Polynomials)
4. Lanczos, C., Applied Analysis, Prentice Hall, 1956.
5. Crandall, S. H., Engineering Analysis, McGraw-Hill 1956.
6. Kantorovich, L. V., and V. I. Krylov, Approximate Methods of Higher Analysis, (C. E. Benster, tran), Interscience, 1964.
7. Multer, R. H., "Exact Nonlinear Model of Wave Generator" ASCE Hyd Div, Jan 1973.
8. Stoker, J. J., Water Waves, Interscience, 1957.

A GENERALIZED RANDOM CHOICE METHOD FOR GAS DYNAMICS

James Glimm*

Guillermo Marshall*

Bradley Flohr

Department of Mathematics
The Rockefeller University
New York, NY 10021

ABSTRACT

We solve a generalization of the Riemann problem for gas dynamical flows influenced by curved geometry, such as flows in a variable-area duct. For this generalized Riemann problem the initial data consists of a pair of steady-state solutions separated by a jump discontinuity. The solution of the generalized Riemann problem is used as a basis for a random choice method in which steady-state solutions are used as an Ansatz to approximate the spatial variation of the solution between grid points. For nearly steady flow in a Laval nozzle, where this Ansatz is appropriate, this generalized random choice method gives greatly improved results.

1. Introduction

Many computational methods for solving gas flow problems are based on approximating the problem with a number of more elementary flow problems, called Riemann problems. The solution of these Riemann problems are important because they provide an explicit and elementary class of solutions which contain extensive information about wave interaction. They are the basic constructive step in the random choice method, and they provide the key input into methods based on front-tracking.

*Also at Courant Institute of Mathematical Sciences,
New York University, New York, NY 10012.

The solutions of Riemann problems for flows influenced by curved geometry exhibit, as characteristic phenomena, a bending and either strengthening or weakening of the waves. Curvature effects arise, for instance, in one-dimensional flows in tubes with variable cross-sectional area and in flows with cylindrical and spherical symmetry. Such flows are called quasi-one-dimensional. Mathematically, the curved geometry introduces a source term in the conservation laws describing the flow, so that the conservation laws are inhomogeneous. This source term influences the speeds and strengths of sound waves and shock waves, so their trajectories are not straight lines when drawn in the space-time plane. The wave speeds and strengths depend on the source term to first order in time, while the wave positions depend on the source terms only to second order in time.

The purpose of this paper is to assess the benefits and difficulties of including second order accuracy in the Riemann problem solution. For this purpose we studied gas flow in Laval nozzles using a generalization of the random choice method. To include second order accuracy, the data for a Riemann problem is inadequate, however. The Riemann problem, a single jump separating two arbitrary constant states, can be thought of as representing a localized portion of a complicated flow field. In order to obtain second order accuracy of the Riemann problem solution it is necessary to give as data not only the value of the states on each side of the jump, but also their spatial derivatives. To do this we suppose that over spatial mesh intervals the solution is a solution of the steady state equations. This gives rise to what we call a generalized Riemann problem, which can be solved to second order in time. Our method of solution incorporates the generalized Riemann problem into the framework of the random choice method; this constitutes what we call the generalized random choice method.

2. The Generalized Random Choice Method

The random choice method is a technique for computing solutions of hyperbolic systems of conservation laws. It consists of approximating the solution at each time step by a piecewise constant state and advancing to the next time step by solving the local Riemann problems formed by the constant states on adjacent mesh intervals. The value of the approximate solution over each mesh interval of the new time step is taken to be the exact solution evaluated at a randomly chosen point. The main advantages of the method lie in its power of resolution for the numerical treatment of discontinuities and sharp interfaces, and in its absence of over- and under-shooting phenomena. The random choice method was introduced by Glimm[3] for homogeneous systems of conservation laws; it was developed into a numerical method by Chorin[1], who made extensive use of it for computations

of combustion problems.

In its present form the random choice method cannot be applied to inhomogeneous hyperbolic systems of conservation laws, such as those describing quasi-one-dimensional gas flows. Several attempts have been made to extend the method to include these problems. Sod[8] developed a straightforward generalization using operator splitting. It consists of a two-step procedure. In the first step the inhomogeneous term is removed and the Riemann problem for the resulting homogeneous system is solved and sampled. In the second step the system of ordinary differential equations obtained by removing the convection terms is solved, using the solution from the first step as initial data. The advantages of this procedure are its simplicity and robustness. However, for certain applications, such as steady nozzle flows, this method requires that the mesh size be quite small to obtain reasonable accuracy.

Another generalization of the random choice method, which uses characteristic tracing, was developed by Marshall and Menendez[6]. This method, by contrast, is a one-step procedure. The Riemann problem for the associated homogeneous system is solved and the influence of the inhomogeneous term is introduced by integration along characteristic curves; only then is the solution sampled. The method of characteristic tracing is more accurate than Sod's splitting method for equal mesh size, but the computational effort for obtaining the same degree of accuracy is greater for characteristic tracing.

In [5] Liu proved global existence for quasi-linear hyperbolic systems, including quasi-one-dimensional gas flow, using a method which generalizes that of Glimm. His results were limited, however, to gas flows which are nowhere sonic (but see [4]). Fok[2] used this method as a basis for constructing a numerical scheme, which he called Liu's scheme. Here the solution at each time step is approximated by a piecewise steady flow. It is advanced to the next time step by solving the ordinary Riemann problems formed by the jumps between steady flow states on adjacent spatial mesh intervals. The approximating steady flow for each mesh interval at the new time step is obtained by sampling this solution at a randomly chosen point (without extrapolation using the steady state equations; cf. the generalized random choice method described below). Fok claims that this method offers only marginal improvement over Sod's method, and only at greater computational cost. In addition, it cannot handle transonic flows.

We now introduce the generalized random choice method. This method is also based on the work of Liu, but is an extension in two respects. Here again the solution at each time step is approximated by a piecewise steady flow. It is

advanced to the next time step by solving, to second order in time, the generalized Riemann problems formed by the steady flows on adjacent spatial mesh intervals. The approximating steady flow for each mesh of the new time step is obtained by sampling this solution at a randomly chosen point and extrapolating from this point by using the steady state equations. Thus we have extended Liu's methods to include the curving of shocks and rarefactions on the level of the local Riemann problem. We have also included a simple stabilizing mechanism in the numerical scheme which allows it to be applied to transonic flows.

The generalized random choice method was applied to transient gas flows, with and without shocks, in a Laval nozzle. We found significant improvement over finite difference methods as well as the above mentioned generalizations of the random choice method. The major reason for this improvement seems to be that the random fluctuations caused by the sampling are greatly reduced: for nearly steady flows the solution is better approximated by piecewise steady flows than by piecewise constant flows.

3. Numerical Results

We present the results of numerical tests using the generalized random choice method applied to the problem of transient gas flows which possesses an asymptotic steady state whose solution is known. We compare these results with those obtained using Sod's splitting method.

In the numerical tests we considered the flow of an inviscid, polytropic, compressible gas through a convergent-divergent (Laval) nozzle. The nozzle, taken from Moretti[7], was composed of four parts, each of length 5.0: an inlet section with constant area 1.5, a sinusoidal contraction to a throat with area 1.0, a sinusoidal expansion from the throat back to area 1.5, and an outlet section of constant area.

The initial conditions for the tests were intended to simulate the starting conditions of a supersonic blow-down tank. A high pressure region occupied the whole nozzle except part of the outlet section, where there was a low pressure region. The boundary conditions were as follows: at the inlet the total temperature and entropy were held constant, while at the outlet the pressure was fixed. The solution of this initial-boundary-value problem consists of transient gas flow which in the large time limit tends to a steady flow with subsonic flow in the inlet, sonic conditions at the throat, and a normal shock downstream of the throat. We describe the transients of the solution by means of contour plots of the pressure in the space-time plane. The asymptotic steady-state solutions are presented in plots of the variation of the Mach number and pressure in space,

superimposed on the exact solution. We remark that because of the presence of sonic conditions at the throat, this problem is particularly difficult for methods based on steady-state solutions.

Figs. 1a, 1b, and 1c present the results up to time 200.0 obtained using the generalized random choice method with 60 grid points. The corresponding results obtained using Sod's method are shown in Figs. 2a, 2b, and 2c. In both methods the general pattern of the transient flow is correctly described: there is a rarefaction wave which travels upstream, partially reflects from the inlet, and finally causes the formation of a stationary shock (which is represented by the closely spaced contour lines). In Sod's method, however, the random fluctuations introduce spurious transients which do not disappear even after long times. These fluctuations diminish the quality of the asymptotic steady-state solutions. In contrast, the generalized random choice method converges to the asymptotic steady-state solution, and the details of the transients appear to be correct. The fluctuations caused by sampling errors are suppressed in this method because of the better approximation of the solution over mesh intervals.

4. Conclusions

We have introduced a generalization of the Riemann problem for gas dynamical flows in a variable-area duct, and have used it as a basis for constructing a generalized random choice method. For nearly steady flows we find this method to be substantially better than other forms of the random choice method and finite difference methods. This is because it reduces fluctuations caused by the random sampling, while maintaining the usual advantages of random choice methods.

References

1. A. J. Chorin, J. Computational Phys. Vol. 23, p.517 (1976).
2. S. K. Fok, Extension of Glimm's Method to the Problem of Gas Flow in a Duct of Variable Cross-section, Thesis, Department of Mathematics, University of California, Berkeley (1981).
3. J. Glimm, Comm. Pure Appl. Math. Vol. 18, p.697 (1965).
4. T. P. Liu, "Transonic Gas Flow in a Duct of Varying Area," Arch. Rat. Mech. Anal., (to appear).

5. T. P. Liu, Comm. Math. Phys. Vol. 68, p.141 (1979).
6. G. Marshall and A. N. Menendez, J. Computational Phys. Vol. 44, p.167 (1981).
7. G. Moretti, Thoughts and Afterthoughts About Shock Computations, Polytechnic Institute of Brooklyn PIBAL Report 72-37 (1972).
8. G. A. Sod, J. Fluid Mech. Vol. 83, p.785 (1977).

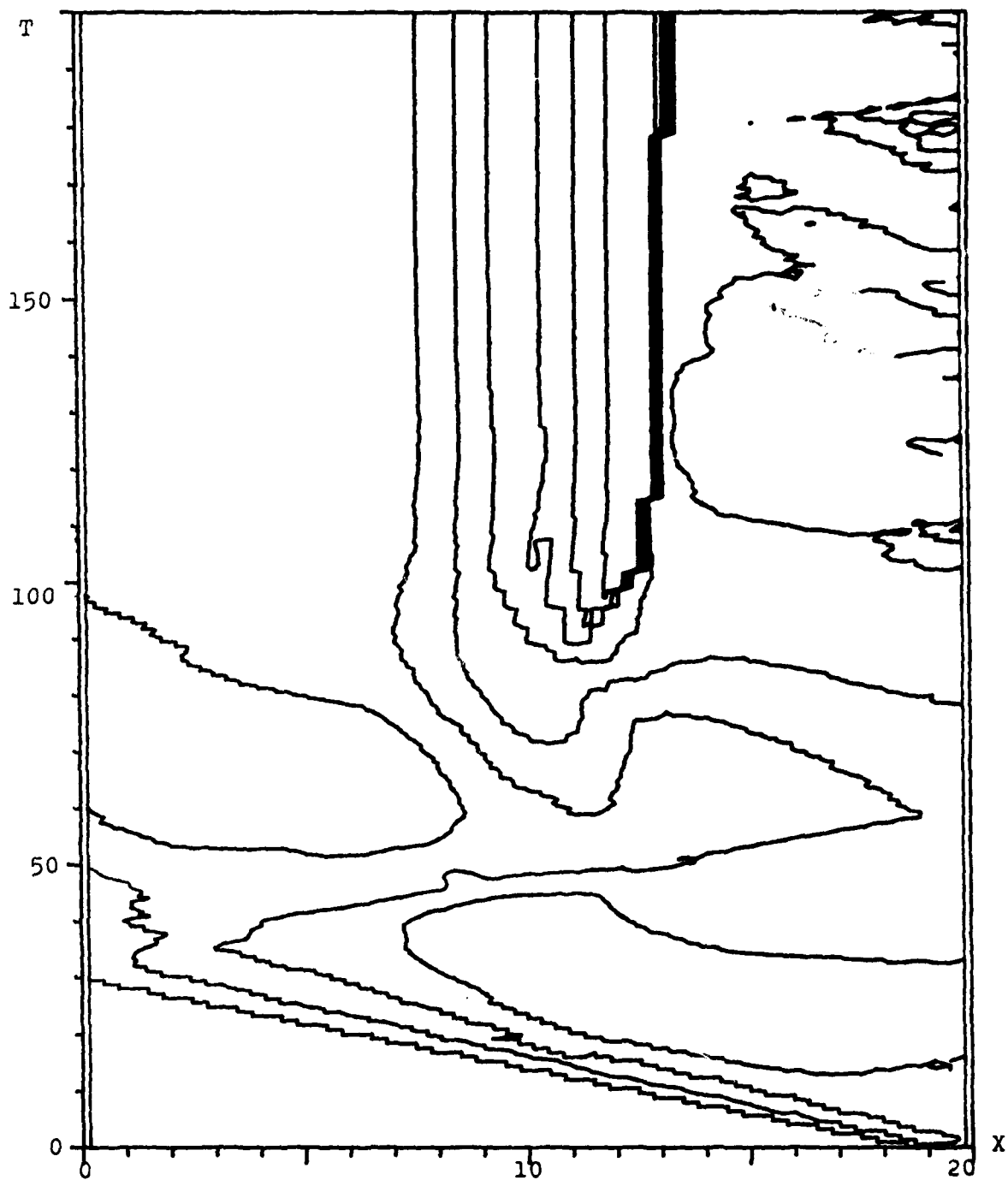


Fig. 1a. Contour plot of the pressure in the space-time plane obtained with the generalized random choice method.

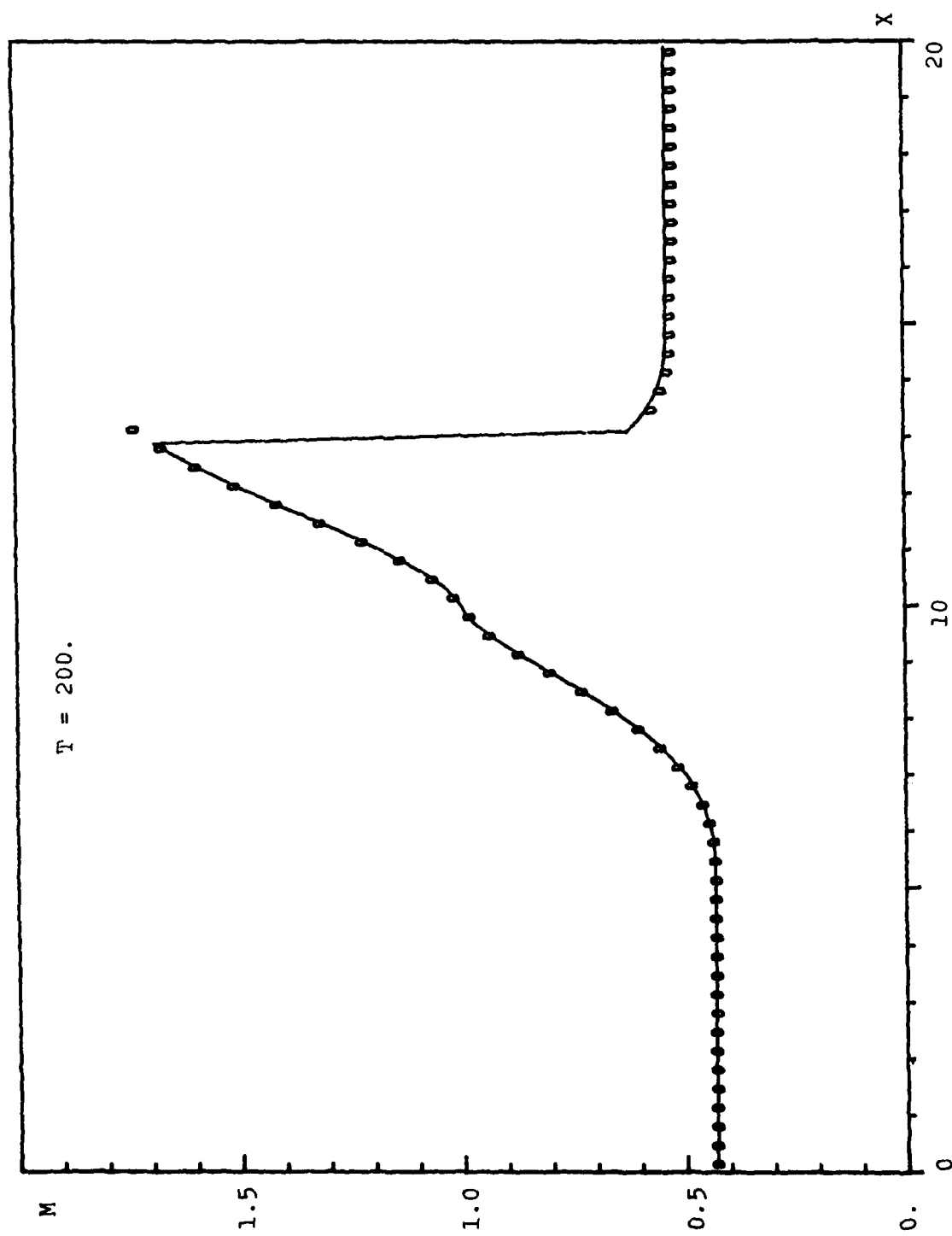


Fig. 1b. Variation of the Mach number in space obtained with the generalized random choice method.

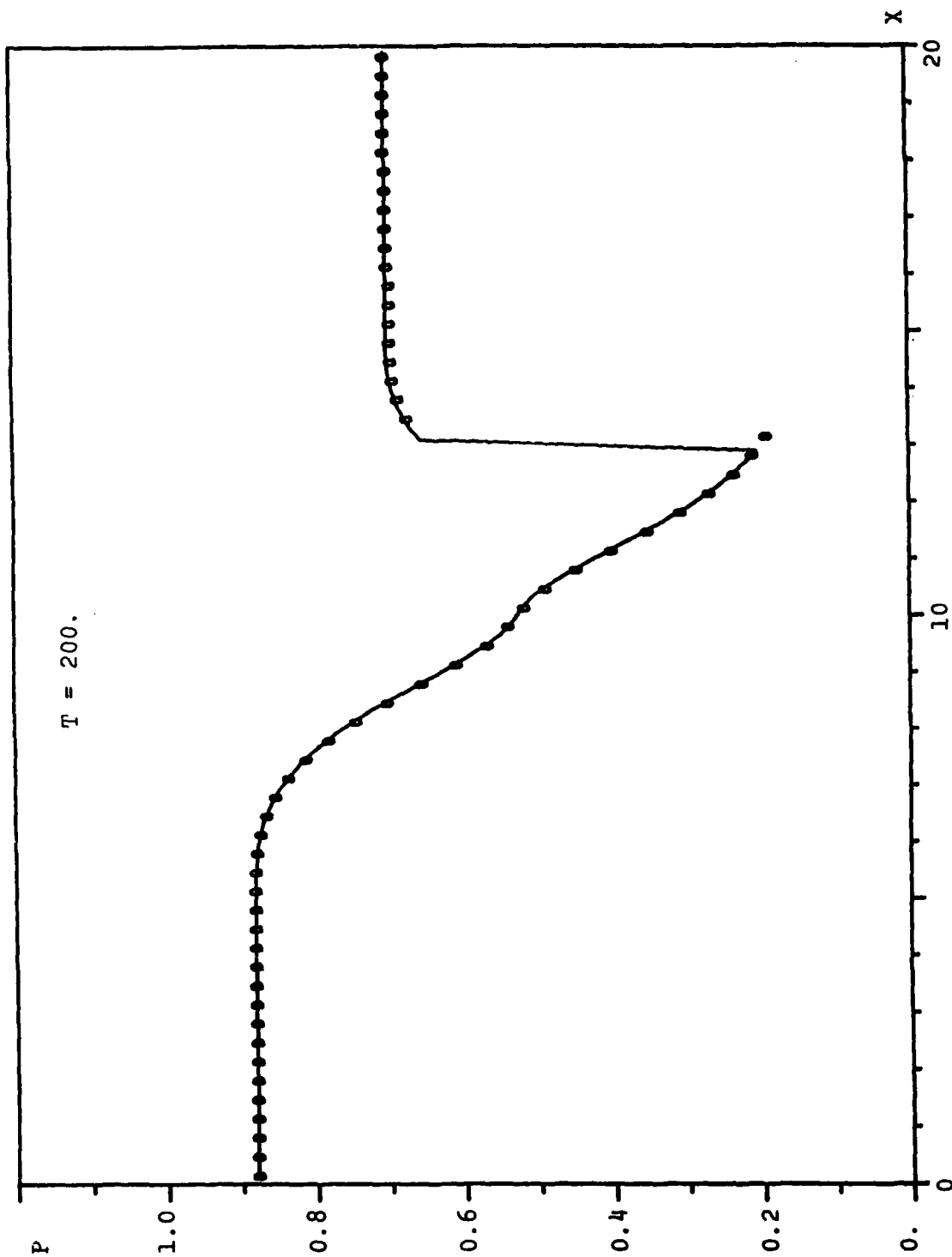


Fig. 1c. Variation of the pressure in space obtained with the generalized random choice method.

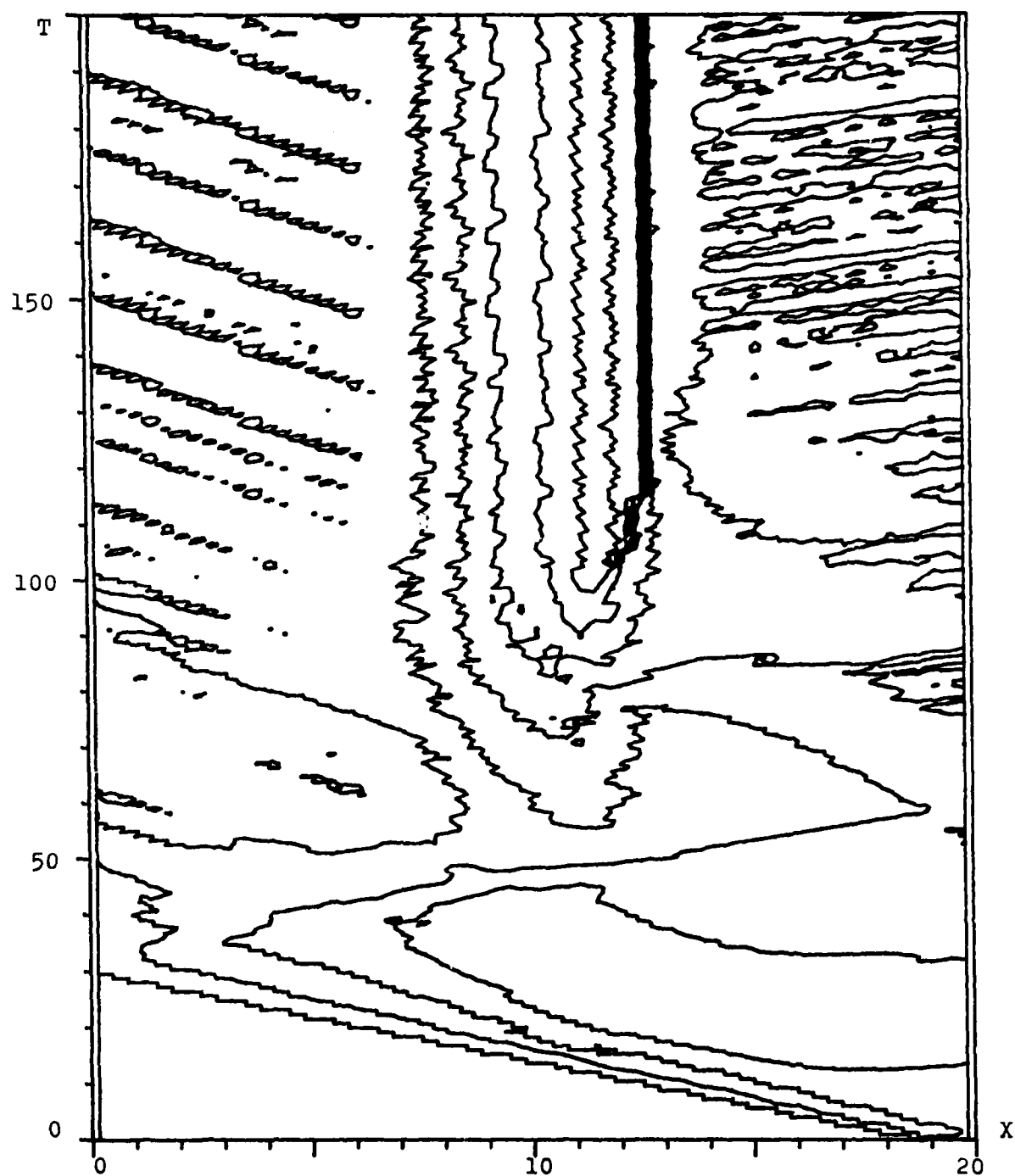


Fig. 2a. Contour plot of the pressure in the space-time plane obtained with Sod's splitting method.

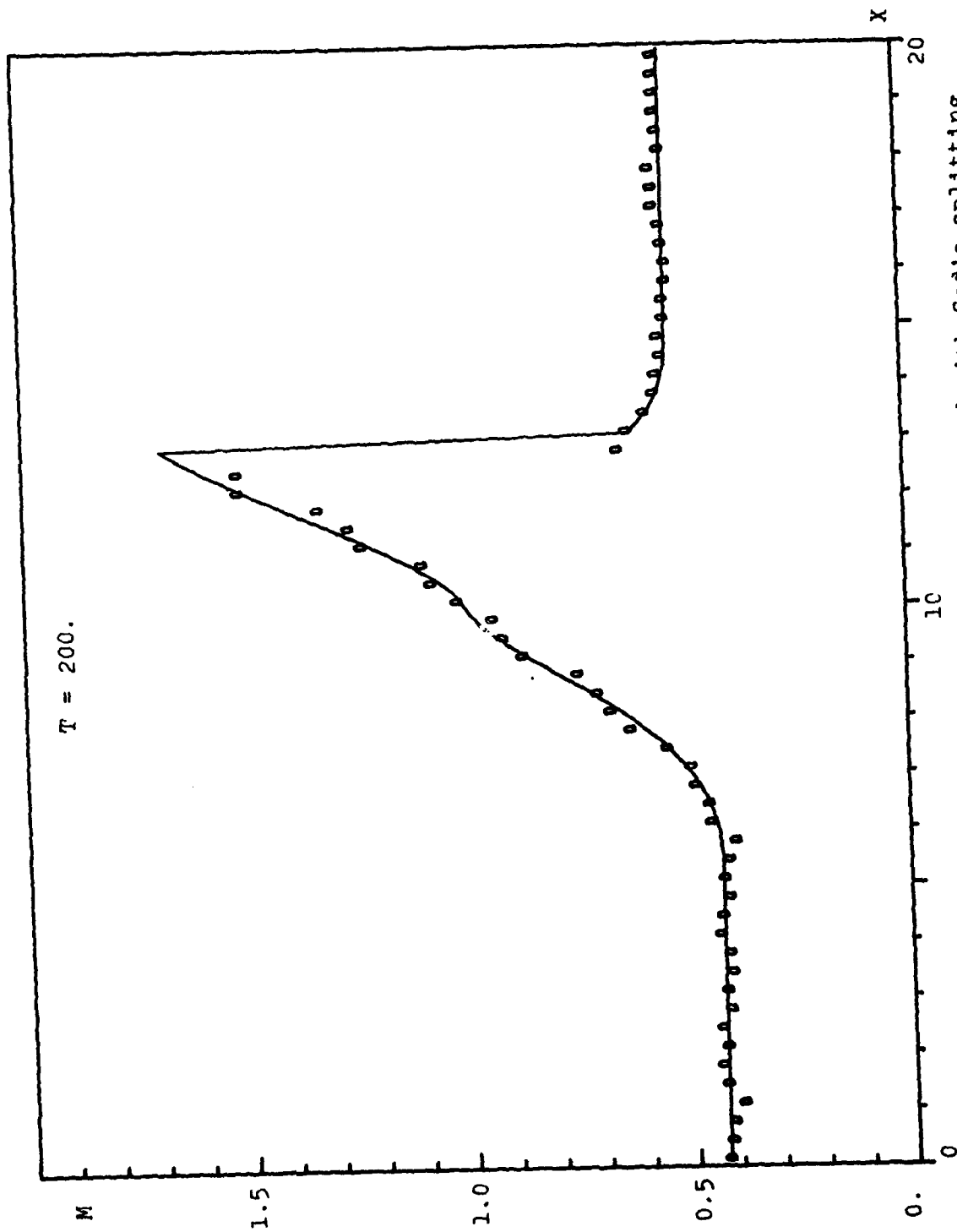


Fig. 2b. Variation of the Mach number in space obtained with Sod's splitting method.

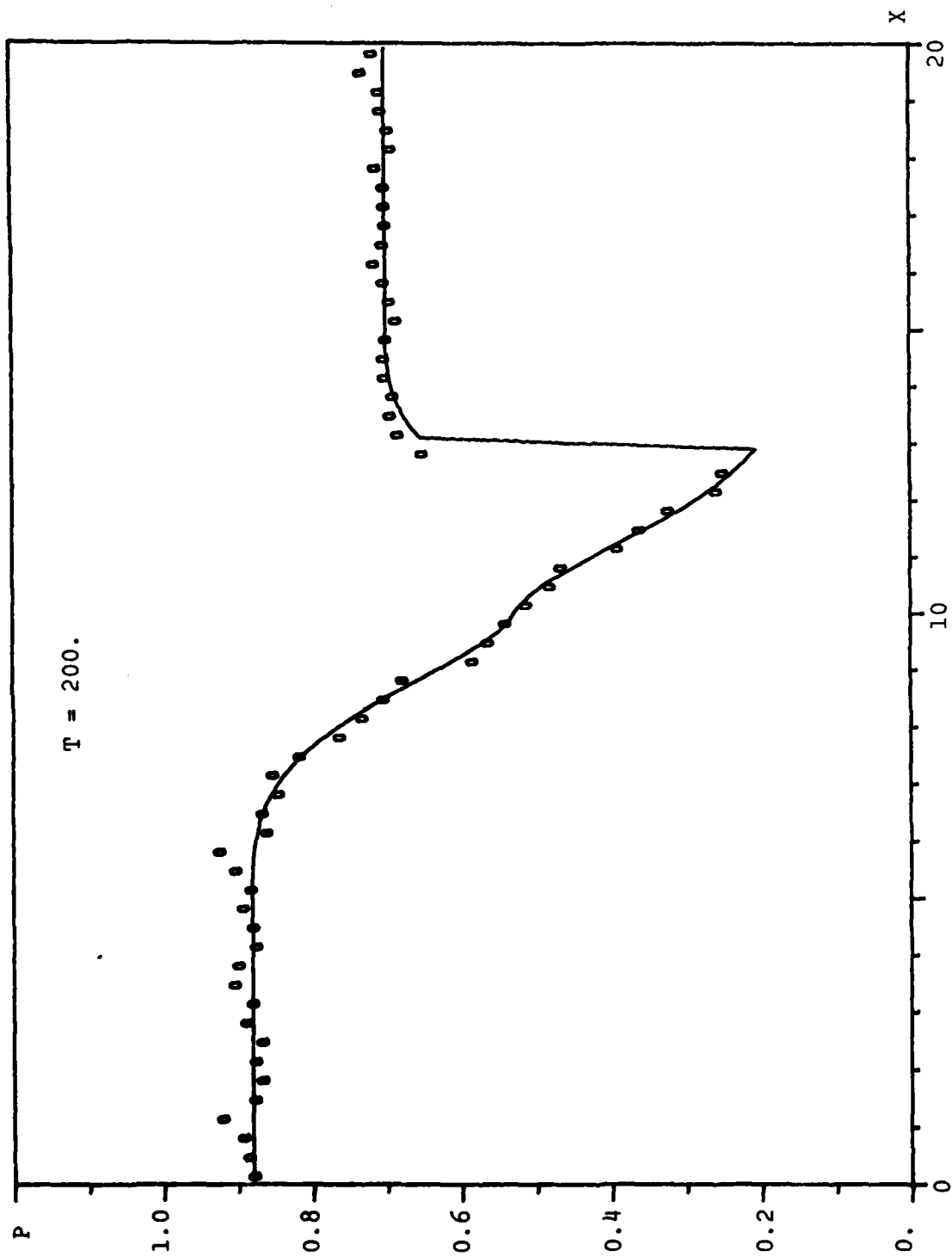


Fig. 2c. Variation of the pressure in space obtained with Sod's splitting method.

**Formulation of Two-Phase Interior Ballistics
Equations for Numerical Treatment**

**Aivars K.R. Celmins
James A. Schmitt**

**Ballistic Research Laboratory
U.S. Army Armament Research and Development Command
Aberdeen Proving Ground, Maryland 21005**

Contents

- 1. Introduction**
- 2. Analytical Basis**
 - 2.1 Assumptions**
 - 2.2 Averaging Integrals and Their Derivatives**
 - 2.2.1 Averaging Volume Integrals**
 - 2.2.2 Time Derivative of Volume Integrals**
 - 2.2.3 Spatial Derivatives of Volume Integrals**
 - 2.2.4 Averaging Surface Integrals**
 - 2.2.5 Differential Equation for Surface Averages**
 - 2.3 Regions of Definition of Averages**
 - 2.4 Averaging Weight Function**
- 3. Conservation Equations**
 - 3.1 Local Equations**
 - 3.1.1 Local Conservation Equations**
 - 3.1.2 Local Interfacial Conditions**
 - 3.2 Averaging of the Local Conservation Equations**
 - 3.2.1 Derivation of the Average Gas Continuity Equation and Porosity Equation**
 - 3.2.2 Derivation of the Average Gas and Solid Momentum Equations**
 - 3.2.3 Derivation of the Average Gas Internal Energy Equation**
 - 3.2.4 Derivation of the Surface Average Equations**
 - 3.3 Summary and Discussion of the Conservation Equations Without Error Terms**
- 4. Governing Equations**
 - 4.1 Basic System of Governing Equations**
 - 4.2 Choice of Dependent Variables**
 - 4.2.1 Particle Number Function**
 - 4.2.2 Pressure Logarithm and Entropy**

- 4.3 Final System of Governing Equations
- 4.4 Regions of Definition
- 4.5 Initial Conditions
- 4.6 Boundary Conditions
- 4.7 Models of Correlations
 - 4.7.1 Equations of State
 - 4.7.2 Acceleration by Gaseous Stresses
 - 4.7.3 Heat Dissipation
 - 4.7.4 Heat Conduction
 - 4.7.5 Acceleration by Drag
 - 4.7.6 Acceleration by Granular Stresses
 - 4.7.7 Burning Rate
 - 4.7.8 Source Terms
 - 4.7.9 Grain Volume and Surface
 - 4.7.10 Grain Surface Heating Rate

5. Summary and Conclusions

References

List of Symbols

Appendix A. Governing Equations for Cylindrically Symmetric Flows in Cylindrical Coordinates

Appendix B. Correlation Model Formulas

1. INTRODUCTION

The flowing medium in a gun tube typically is a mixture of a compressible gas with burning solid propellant grains. Details of the flow are important for weapons development, but only bulk properties can be routinely measured, such as the trajectory of the projectile, the pressure history at a fixed station, the heating of the gun tube, etc. Therefore, a need exists for a detailed mathematical model of interior ballistics two-phase flows.

A complete mathematical description of the flow could provide the motion and combustion history of each propellant grain, and of the gas flow between the grains. The corresponding local governing equations are easily established, but they cannot be solved numerically because of the great number of grid points needed to describe a flow with many moving interfaces. The computational work can be reduced only by sacrificing the detailed description of the flow. To that end one considers mean values of the two-phase flow that are derived from the local properties of the gas and grains. The governing equations for these average properties are established by averaging the local governing equations.

In the present paper we derive the governing equations for a particular set of averages. The averaging process is chosen with the special needs of interior ballistics in mind and with attention to the numerical solution of the ensuing equations.

Previous work on two-phase equations for interior ballistics has been done by Gough (1974), Kuo et al. (1976), Fisher and Trippe (1974), and Krier et al. (1974). Gough's equations were later augmented and used in a computer program developed by Gibeling et al. (1980). Our equations are different because we have used a different averaging process, chosen a different set of dependent variables, and changed some correlation models that provide experimental input to the theory.

The averages in this report are computed by weighted averaging over a finite volume. Gough (1974) used instead a weighted averaging over an infinite space-time domain with an unspecified weight function. The rationale of our choice is based on the observation that any averaging smooths out local details. In order not to lose too many details, one should, therefore, use the smallest averaging domain that is compatible with the requirements of the problem at hand. One requirement of the averages is that they should be differentiable as many times as the ensuing governing equations indicate. It has been shown by Delhaye and Achard (1977) that line or surface averages of a gas/particle mixture do not possess the required differentiability properties. Therefore, the smallest domain for averaging is a three-dimensional volume. Time averaging is not needed to insure differentiability, if the weight function for space averaging is chosen properly (see Section 2.2). If one, nevertheless, chooses to time average, then the time average interval would have to be very small because we are interested in an accurate characterization of a rapidly changing flow field.

The size of the averaging volume is important. The use of an infinite volume for averaging is not appropriate in confined flows because the sum of the volume fractions of the two phases is not equal to one. This creates problems for the formulation of the governing equations and the boundary conditions, and for the interpretation of the results. The problem with the formulation of the equations is eliminated by using an appropriate finite volume average, while the others become more easily tractable. We discuss the problems in Sections 4.4 and 4.6.

The average equations which are derived in Section 3 include the effects of viscosity of the gas and of turbulence. Furthermore, the choice of equations for averaging and the choice of dependent variables has a bearing on the numerical solution of the equations. We have chosen a set of variables that eliminates some possible numerical singularities, enhances the accuracy of numerical differentiation, and separates important physical processes for easier modeling. The choice of variables is discussed in Section 4.2. We also have chosen the internal energy equation for averaging instead of the commonly used total energy equation. The reasons for this choice are that it produces a clear separation of physical effects and a more lucid modeling of two-phase phenomena. They are discussed in Sections 3.2.3 and 4.7.3, respectively. As a result of the considerations of viscous effects and the above choices, our governing equations differ from those derived by Gough. Each set of equations has different approximation errors and requires some different models of experimental correlations.

The experimental correlations in interior ballistics are characterized by a scarcity of data. This is one reason why corresponding mathematical models have not been firmly established. In Section 4.7 we list a set of correlations, most of which are based on Gough's work. Some improvements and changes reflect the difference of our approach.

Even with the reduction of the problem size by the change from local to average functions, one is faced with a formidable numerical problem. Typically, in a two-phase flow one has a set of eleven non-linear partial differential equations. (Thirteen equations if a turbulence model is included.) In order to describe the three-dimensional flow in reasonable detail one has to specify the eleven variables at a minimum of about 54,000 grid points. Therefore, whenever possible, one would exploit the cylindrical symmetry of the gun. If also the flow is cylindrically symmetric, then the number of grid points may be reduced to about 1,500. The proper coordinates for flows with cylinder symmetry are cylinder coordinates and we have, therefore, listed in Appendix A all equations in cylinder coordinates, thereby also assuming that the flow is independent of the circumferential coordinate.

2. ANALYTICAL BASIS

2.1 Assumptions

In the next three Sections (2.2, 2.3, and 2.4) we shall discuss some properties of averaged functions and develop general formulas that are needed for the derivations in Section 3. The averages to be discussed are weighted space averages over a finite averaging volume. We do not try to establish general properties of such averages but rather concentrate on what is needed for a specific interior ballistics modeling. For that application, the quantities to be averaged are the local properties of a gas and of propellant particles within the averaging volume. We assume that no other material is present in the tube.

The gas is assumed to be non-reacting and obeying algebraic equations of state, that permits one to express all thermodynamic variables in terms of two such quantities. The particular equations of state considered are the Noble-Abel equation with a constant ratio of specific heats. However, most of the results are independent of the particular equations of state.

We will assume that the gas is in a state without shocks within the averaging volume. This is necessary to have average equations with the proper differentiability conditions. Particular differentiability conditions of the local gas properties will be enumerated in Section 2.2.

If shocks are present in the gas flow, then one could average only over the shock free regions and treat the shocks as explicit boundaries. However, this approach has serious drawbacks because of the uncertainty of the corresponding boundary conditions (see Section 4.6). Space or time averaging is not the appropriate technique for the treatment of interior ballistics flows with shocks or other internal discontinuities.

The propellant particles are assumed to be incompressible and elastic. However, we shall neglect all effects of the rotation of the solid particles. Like in the gas, the local material properties within and on the surface of each particle are assumed to be differentiable functions of time and space. Particulars of the differentiability conditions will be enumerated in Section 2.2.

2.2 Averaging Integrals and Their Derivatives

2.2.1 Averaging Volume Integrals. We define the averaging volume $V(x)$ as the inside of a closed surface $S(x)$. Both are independent of time and dependent on a spacial coordinate vector x as a parameter. For instance, if $V(x)$ is a sphere, then x may be chosen as the center of the sphere. About the surface $S(x)$, we assume that it has a well defined normal almost everywhere. The shape and the size of the averaging volume are assumed to be constant.

The particles are defined by corresponding surfaces, s_{pi} . Because the particles are moving and burning, the s_{pi} are functions of time, but they are independent of the parametric coordinate vector x . We assume that the particle surfaces, too, have well defined normals almost everywhere. We define as S_p the union of all those particle surfaces s_{pi} that are within the averaging volume V , including its surface S_v . Accordingly, the intersection $S \cap S_v$ can have a finite area. Most often, the area of the intersection will be zero (Figure 1).

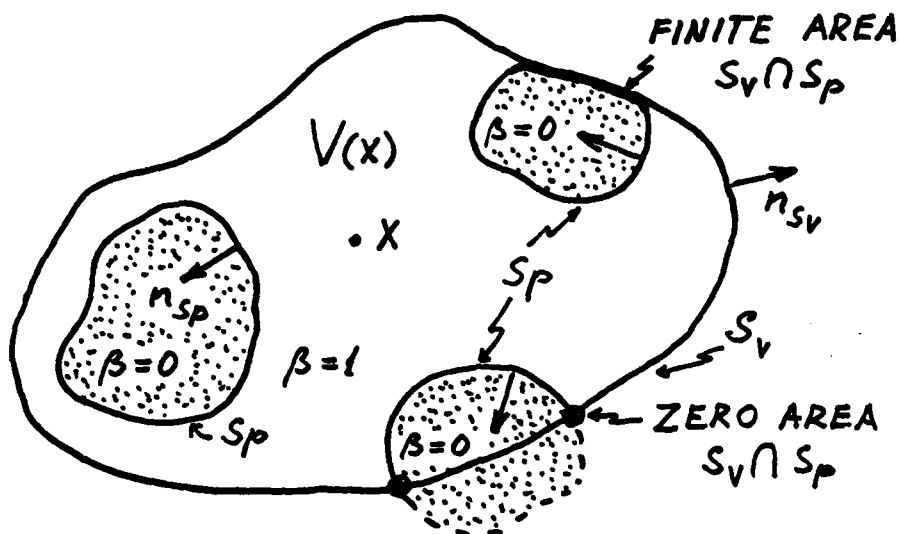


Figure 1. Averaging Volume

All averages will be defined by integrals over the space occupied either by gas or by particles. In order to have a convenient notation for the corresponding integrals, we define a phasic function β as follows

$$\beta(t, \xi) = \begin{cases} 0 & \text{if } \xi \text{ is inside a particle at time } t \\ 1 & \text{if } \xi \text{ is outside particles or on a particle surface at time } t \end{cases} \quad (2.1)$$

We will also use a non-negative weight function g for the calculation of averages. Let

$$VG = \int_{V(x)} g(\xi-x) dV(\xi) = \text{constant} \quad (2.2)$$

be the integral of the weight function ("the weighted averaging volume"). Then the weighted volume fraction occupied by gas is

$$\alpha(t,x) = \frac{1}{VG} \int_{V_{\text{gas}}(t,x)} g(\xi-x) dV(\xi) = \frac{1}{VG} \int_{V(x)} \beta(t,\xi) g(\xi-x) dV(\xi) . \quad (2.3)$$

The intrinsic average $\phi(t,x)$ of a function $\check{\phi}(t,x)$ that is defined in the regions occupied by gas is defined by

$$\begin{aligned} \alpha(t,x) \phi(t,x) &= \frac{1}{VG} \int_{V_{\text{gas}}(t,x)} g(\xi-x) \check{\phi}(t,\xi) dV(\xi) \\ &= \frac{1}{VG} \int_{V(x)} \beta(t,\xi) g(\xi-x) \check{\phi}(t,\xi) dV(\xi) . \end{aligned} \quad (2.4)$$

Notice that, whereas $\check{\phi}(t,x)$ is defined only within regions occupied by gas, the average $\phi(t,x)$ is defined for all values of x (within limits outlined in Section 2.3).

A corresponding average $\phi^*(t,x)$ of a function $\check{\phi}^*(t,x)$ that is defined only inside the particles is given by

$$[1-\alpha(t,x)]\phi^*(t,x) = \frac{1}{VG} \int_{V(x)} [1-\beta(t,\xi)] g(\xi-x) \check{\phi}^*(t,\xi) dV(\xi) . \quad (2.5)$$

Sufficient conditions for the existence of the average function are the piecewise continuity with respect to x of the functions $\check{\phi}(t,x)$ and $\check{\phi}^*(t,x)$ within their regions of definition. Obviously, the average of any function of time only is the function itself.

2.2.2 Time Derivative of Volume Integrals. The averaging integrals (2.3), (2.4), and (2.5) define functions of t and x . In this section we formulate differentiability conditions of the average functions with respect to time t .

Applying Leibnitz formula (Truesdell and Toupin, 1960) to an averaging integral (2.4) over V_{gas} we obtain

$$\begin{aligned} \frac{\partial}{\partial t} \int_{V_{\text{gas}}(t,x)} \psi(t,x,\xi) dV(\xi) = & \int_{V_{\text{gas}}(t,x)} \frac{\partial}{\partial t} [\psi(t,x,\xi)] dV(\xi) + \\ & + \int_{S_p(t,x)} [\psi(t,x,\xi(\eta)) (u_{sp} \cdot n_{sp})] dS(\eta) , \end{aligned}$$

or

(2.6)

$$\begin{aligned} \frac{\partial}{\partial t} \int_{V(x)} \beta(t,\xi) \psi(t,x,\xi) dV(\xi) = \\ = \int_{V(x)} \beta \frac{\partial}{\partial t} \psi dV(\xi) + \int_{S_p(t,x)} \psi(t,x,\xi(\eta)) (u_{sp} \cdot n_{sp}) dS(\eta) , \end{aligned}$$

where u_{sp} is the velocity of a point of S_p and n_{sp} is the outward unit normal of S_p at the same point. (The "outward" normal points by definition into the grains, Figure 1.) The surface integral is only over S_p and not over S_v because the latter surface is assumed to be stationary.

The first integral on the right-hand side of Eq. (2.6) exists and is a continuous function of x and t if $\partial\psi/\partial t$ is a continuous function of x and t , and a piecewise continuous function of ξ . The surface integral over S_p in Eq. (2.6) exists if the surface velocity is finite. However, the area of the surface S_p has discontinuities with respect to x and, possibly, with respect to t , whenever the intersection $S_p \cap S_v$ has a finite area. Therefore, the surface integral is a continuous function of x and t only if $\psi = 0$ on S_v .

Because in our case

$$\psi(t,x,\xi) = g(\xi-x) \tilde{\phi}(t,\xi) , \quad (2.7)$$

we may formulate the following sufficient conditions for the continuity of the time derivative of the averaging integral in terms of g and $\tilde{\phi}$:

$$\left.
\begin{aligned}
&\frac{\partial \tilde{\phi}(t, \xi)}{\partial t} \quad \text{is continuous with respect to } t \text{ and piecewise} \\
&\quad \text{continuous with respect to } \xi, \\
&g(\xi-x) = 0 \quad \text{on the surface } S_v, \\
&g(\xi-x) \quad \text{is continuous in } V.
\end{aligned}
\right\} (2.8)$$

We notice that the condition on $\tilde{\phi}$, of course, applies only to the regions where $\tilde{\phi}$ is defined.

The differentiation formula (2.6) is in terms of g and $\tilde{\phi}$

$$\int_V \beta g \frac{\partial \tilde{\phi}(t, \xi)}{\partial t} dV(\xi) = \frac{\partial}{\partial t} \int_V \beta g \tilde{\phi} dV - \int_{S_p} g \tilde{\phi} (u_{sp} \cdot n_{sp}) dS. \quad (2.9)$$

The corresponding formula for functions $\tilde{\phi}^*$ that are defined within the solid grains is

$$\int_V (1-\beta) g \frac{\partial \tilde{\phi}^*}{\partial t} dV = \frac{\partial}{\partial t} \int_V (1-\beta) g \tilde{\phi}^* dV + \int_{S_p} g \tilde{\phi}^* (u_{sp} \cdot n_{sp}) dS. \quad (2.10)$$

In the latter formula, the surface normal n_{sp} again points into the grains. Because now we are integrating over the inside of the grains, the sign of the last integral in Eq. (2.10) is opposite to that of the corresponding integral in Eq. (2.9).

2.2.3 Spacial Derivatives of Volume Integrals. Applying Leibnitz type formula to an averaging integral (2.4) over V_{gas} one obtains*

$$\nabla_x \int_{V(x)} \beta(t, \xi) \psi(t, x, \xi) dV(\xi) = \int_V \beta \nabla_x \psi dV + \int_{S_v - S_p} \psi n_s dS + \int_{S_v \cap S_p} \psi n_{sp} dS. \quad (2.11)$$

*We note that ψ could be a scalar, a vector, or a second order tensor. For example, if ψ is a vector, a dot signifying the divergence of ψ and the dot product of ψ with the normal should be used in Eq. (2.11). For simplicity, the use of the dot is omitted in Section 2 wherever ψ is not specified. The understood presence or absence of the dot should be clear from the context.

Gauss theorem (Fulks, 1969, p. 354) applied to the same integration volume is

$$\int_V \beta \nabla_{\xi} \psi dV = \int_{S_V - S_P} \psi n_s dS + \int_{S_P} \psi n_{sp} dS . \quad (2.12)$$

Subtracting Eq. (2.12) from Eq. (2.11) one obtains

$$\nabla_x \int_V \beta \psi dV = \int_V \beta (\nabla_x + \nabla_{\xi}) \psi dV - \int_{S_P} \psi n_{sp} dS + \int_{S_P \cap S_V} \psi n_{sp} dS . \quad (2.13)$$

Sufficient conditions for the continuity of the right-hand side of Eq. (2.13) are

$$\left. \begin{aligned} (\nabla_x + \nabla_{\xi}) \psi(t, x, \xi) & \text{ is continuous with respect to } x \text{ and } t, \text{ and} \\ & \text{ piecewise continuous with respect to } \xi, \\ \psi = 0 & \text{ on } S_V . \end{aligned} \right\} \quad (2.14)$$

In our application we want some of the average functions to be differentiable twice with respect to the spacial variables. By a formal differentiation of Eq. (2.13) we obtain, assuming that $\psi = 0$ on S_V ,

$$\nabla_x \nabla_x \int_V \beta \psi dV = \nabla_x \int_V \beta (\nabla_x + \nabla_{\xi}) \psi dV - \nabla_x \int_{S_P} \psi n_{sp} dS . \quad (2.15)$$

Next, we apply the formula (2.13) to the first integral on the right-hand side of Eq. (2.15) obtaining

$$\begin{aligned} \nabla_x \int_V \beta (\nabla_x + \nabla_{\xi}) \psi dV &= \int_V \beta (\nabla_x + \nabla_{\xi}) (\nabla_x + \nabla_{\xi}) \psi dV - \\ & - \int_{S_P} (\nabla_x + \nabla_{\xi}) \psi n_{sp} dS + \int_{S_P \cap S_V} (\nabla_x + \nabla_{\xi}) \psi n_{sp} dS . \end{aligned} \quad (2.16)$$

The surface integral in (2.15) is

$$\nabla_x \int_{S_P} \psi n_{sp} dS = \int_{S_P} \nabla_x \psi n_{sp} dS . \quad (2.17)$$

Sufficient continuity conditions for (2.16) are

$$\left. \begin{aligned} (\nabla_{\mathbf{x}} + \nabla_{\xi})(\nabla_{\mathbf{x}} + \nabla_{\xi})\psi & \text{ is piecewise continuous with respect to } \xi, \text{ and continuous with respect to } t \text{ and } \mathbf{x}, \\ (\nabla_{\mathbf{x}} + \nabla_{\xi})\psi & = 0 \quad \text{on } S_v. \end{aligned} \right\} (2.18)$$

Sufficient for the continuity of (2.17) is that

$$\nabla_{\mathbf{x}}\psi \quad \text{is continuous with respect to } t \text{ and } \mathbf{x} \text{ and piecewise continuous with respect to } \xi. \quad (2.19)$$

Because $\psi(t, \mathbf{x}, \xi) = g(\xi - \mathbf{x})\tilde{\phi}(t, \xi)$, we may express the continuity conditions in terms of $g(\xi - \mathbf{x})$ and $\tilde{\phi}(t, \xi)$ as follows.

Sufficient for the continuity of first order spacial derivatives of the averaging integral is that (see Eq. (2.14))

$$\left. \begin{aligned} \nabla_{\xi}\tilde{\phi}(t, \xi) & \text{ is piecewise continuous with respect to } \xi \text{ and continuous with respect to } t, \\ g(\xi - \mathbf{x}) & \text{ is continuous,} \\ g(\xi - \mathbf{x}) & = 0 \quad \text{on } S_v. \end{aligned} \right\} (2.20)$$

The integration formula (2.13) in terms of g and $\tilde{\phi}$, if the conditions (2.20) are satisfied, is

$$\int_{V(\mathbf{x})} g \nabla_{\xi}\tilde{\phi}(t, \xi) dV(\xi) = \nabla_{\mathbf{x}} \int_{V} g \tilde{\phi} dV + \int_{S_p} g \tilde{\phi} n_{sp} dS. \quad (2.21)$$

The corresponding formula to (2.21) for functions $\tilde{\phi}$ defined within the solid grains is

$$\int_{V(\mathbf{x})} [1-\beta] g \nabla_{\xi} \tilde{\phi}(t, \xi) dV(\xi) = \nabla_{\mathbf{x}} \int_{V(\mathbf{x})} [1-\beta] g \tilde{\phi} dV - \int_{S_p} g \tilde{\phi} n_{sp} dS. \quad (2.22)$$

The continuity conditions (2.18) and (2.19) for second order derivatives are in terms of g and $\tilde{\phi}$ as follows

$$\left. \begin{aligned} \nabla_{\xi} \nabla_{\xi} \tilde{\phi}(t, \xi) & \text{ is piecewise continuous with respect to } \xi \\ & \text{ and continuous with respect to } t, \\ \nabla_x g(\xi-x) & \text{ is piecewise continuous (This suffices because} \\ & \tilde{\phi} \text{ is continuously differentiable due to the} \\ & \text{ first condition, Eq. (2.20)) ,} \end{aligned} \right\} (2.23)$$

$$g(\xi-x) = 0 \quad \text{on } S_v .$$

The integration formula (2.15) is, if these conditions are satisfied,

$$\int_{V(x)} \beta g \nabla_{\xi} \nabla_{\xi} \tilde{\phi}(t, \xi) dV(\xi) = \nabla_x \nabla_x \int_{V(x)} \beta g \tilde{\phi} dV + \int_{S_p} g \nabla_{\xi} \tilde{\phi} n_{sp} dS + \int_{S_p} (\nabla_x g) \tilde{\phi} n_{sp} dS . \quad (2.24)$$

In summary, if the weight function g is chosen such that its first derivatives are piecewise continuous, $g > 0$ in V , and $g = 0$ on S_v , then the averaging integrals are continuously differentiable at least once if $\tilde{\phi}$ is differentiable, and at least twice if $\tilde{\phi}$ is twice differentiable within its region of definition.

2.2.4 Averaging Surface Integrals. Some flow properties are only defined on the surface of the propellant grains, e.g., the burning rate, the regression distance, and the surface temperature. The corresponding averages are computed by surface integrals.

The weighted area of the grain surface that is contained in the averaging volume is

$$SG = \int_{S_p(t,x)} g(s(t,\eta)-x) dS(\eta) , \quad (2.25)$$

where $s(t,\eta)$ defines the surface and η represents surface coordinates. Contrary to the weighted averaging volume VG , the surface SG is not a constant but a function of t and x .

Average surface functions are defined by

$$\phi(t,x) = \frac{1}{SG} \int_{S_p(t,x)} g(s(t,\eta)-x) \tilde{\phi}(t,\eta) dS(\eta) . \quad (2.26)$$

We discuss the differentiability of the surface averages by considering a single grain. Let its surface $s(t, \eta)$ be defined in Cartesian coordinates by

$$s(t, \eta) = \begin{pmatrix} x_s(t, \eta) \\ y_s(t, \eta) \\ z_s(t, \eta) \end{pmatrix}. \quad (2.27)$$

Then the surface element $dS(\eta)$ is defined by (Courant and John, 1974)

$$dS = Z(t, \eta) d\eta, \quad (2.28)$$

where $d\eta$ is the product of the differentials of the components of η , $Z(t, \eta) = \left(\det \left[\left(\frac{\partial s}{\partial \eta} \right)^T \left(\frac{\partial s}{\partial \eta} \right) \right] \right)^{1/2}$, and $\partial s / \partial \eta$ is the Jacobian matrix of the function $s(t, \eta)$.

The contribution of the grain to the weighted grain surface is according to Eq. (2.25)

$$SG_1 = \int_{\eta_1}^{\eta_2} g(s(t, \eta) - x) Z(t, \eta) d\eta. \quad (2.29)$$

The time derivative of SG_1 is

$$\frac{\partial}{\partial t}(SG_1) = \int_s (-\nabla_x g) \cdot \frac{\partial s}{\partial t} Z d\eta + \int_s g \frac{\partial Z}{\partial t} d\eta + \left[\int_{s \cap S_v} g Z dC \right] \frac{\partial C}{\partial t}. \quad (2.30)$$

The integral in the last term in Eq. (2.30) is to be taken over the intersection C of the grain surface s with the boundary S_v of the averaging volume. If we assume that $g = 0$ on S_v then the integral is identically zero, and we do not have to specify conditions for $\partial C / \partial t$.

Sufficient conditions for the right-hand side of Eq. (2.30) to be a continuous function of x and t are

$$\frac{\partial^2 s}{\partial \eta \partial t}$$

is piecewise continuous with respect to η
and continuous with respect to t ,

$$g = 0$$

on S_v ,

(2.31)

$$\nabla_x g$$

is continuous, with possible exception of
isolated singular points ,

$$\nabla_x g = 0$$

on S_v .

The first condition in Eq. (2.31) is satisfied if the grain surface has almost everywhere a normal. The next two conditions on $g(\xi-x)$ are essentially the same as encountered before in the discussion of volume averages. The last condition on g is new, and it needs to be introduced if $\partial s/\partial t$ is not equal to zero and the intersection $s \cap S_v$ has a finite area. (See the comment to Eq. (2.6).)

Next, we consider the spacial derivatives of SG_1 . One obtains according to Leibnitz type rule

$$\nabla_x (SG_1) = \int_s \nabla_x g Z d\eta + \left[\int_{s \cap S_v} g Z dC \right] \frac{\partial C}{\partial x} . \quad (2.32)$$

The right-hand side of Eq. (2.32) obviously is continuous if the conditions (2.31) are satisfied.

If the averaging volume contains several grains then SG is the sum of the individual SG_1 . The sum is continuously differentiable if each of the grains satisfies the first condition in Eq. (2.31), and g satisfies the other three conditions.

We now turn to the surface average function $\phi(t,x)$, defined by Eq. (2.26). We notice that ϕ is a continuous function of all its arguments, if the conditions (2.31) are satisfied and the surface function $\tilde{\phi}(t,\eta)$ is continuous with respect to time and piecewise continuous with respect to η . We assume that $\tilde{\phi}$ possess these properties and reformulate Eq. (2.26) as follows

$$\phi \Sigma(SG_1) = \Sigma \left(\int_{s_{p1}} g \tilde{\phi} dS \right). \quad (2.33)$$

The time derivative of the left-hand side of Eq. (2.33) is

$$L_t = \phi \frac{\partial}{\partial t} (SG) + (SG) \frac{\partial \phi}{\partial t} . \quad (2.34)$$

The first term in this expression is continuous under our assumptions. Therefore, also the second term (and $\partial \phi / \partial t$) is continuous, if the time derivative of the right-hand side of Eq. (2.33) is continuous. The contribution of each term on the right-hand side of Eq. (2.33) to the time derivative is, via Eq. (2.30)

$$\begin{aligned} R_{t1} = & \int_{s_{p1}} (-\nabla_x g) \frac{\partial g}{\partial t} \tilde{\phi} Z \, d\eta + \int_{s_{p1}} g \frac{\partial \tilde{\phi}}{\partial t} Z \, d\eta + \\ & + \int_{s_{p1}} g \tilde{\phi} \frac{\partial Z}{\partial t} \, d\eta + \left[\int_{s_{p1} \cap S_v} g \tilde{\phi} Z \, dC \right] \frac{\partial C}{\partial t} . \end{aligned} \quad (2.35)$$

R_{t1} is a continuous function of x and t if in addition to the condition (2.31) $\tilde{\phi}$ also satisfies the condition

$$\frac{\partial \tilde{\phi}}{\partial t} \quad \text{is a continuous function of } t \text{ and a piecewise continuous function of } \eta. \quad (2.36)$$

Because ϕ and $(SG)_t$, in Eq. (2.34), are continuous functions if (2.31) and (2.36) are satisfied, these conditions are sufficient to insure that $\phi(t, x)$ is continuously differentiable with respect to time.

In order to investigate the spacial differentiability of $\phi(t, x)$ we differentiate Eq. (2.33) with respect to x . On the left-hand side we obtain

$$L_x = \phi \nabla_x (SG) + (SG) \nabla_x \phi . \quad (2.37)$$

On the right-hand side of Eq. (2.33), each summand produces the expression

$$R_{x1} = \int_{s_{p1}} (\nabla_x g) \tilde{\phi} Z \, d\eta + \left[\int_{s_{p1} \cap S_v} g \tilde{\phi} Z \, dC \right] \nabla_x C . \quad (2.38)$$

R_{x1} is continuous if the conditions (2.31) are satisfied. Because $\phi \nabla_x (SG)$ is continuous, the conditions are sufficient for continuous differentiability of ϕ with respect to the spacial coordinate.

Second order spacial derivatives of surface averaged quantities do not enter the governing equations. Therefore, we do not formulate existence conditions for these derivatives.

2.2.5 Differential Equation for Surface Averages. All surface averages satisfy a differential equation for material properties. We shall derive the equation in this section.

Let $U(t, x)$ be an arbitrary velocity vector and let g satisfy the conditions (2.31). Then one can combine Eqs. (2.30) and (2.32) obtaining for the sum SG of all individual SG_i .

$$\frac{\partial}{\partial t} (SG) + U \nabla_x (SG) = \int_{S_p} \nabla_x g \cdot (U - \frac{\partial s}{\partial t}) Z \, d\eta + \int_{S_p} g \frac{\partial Z}{\partial t} \, d\eta \quad (2.39)$$

The integrals on the right-hand side are taken over S_p , i.e., over all grain surfaces contained in the averaging volume.

A corresponding formula can be derived for the product $(SG) \phi$ from Eqs. (2.34), (2.35), (2.37), and (2.38) with the result

$$\begin{aligned} \frac{\partial}{\partial t} ((SG) \phi) + U \nabla_x ((SG) \phi) &= \int_{S_p} (\nabla_x g) \cdot (U - \frac{\partial s}{\partial t}) \tilde{\phi} Z \, d\eta + \\ &+ \int_{S_p} g \tilde{\phi} \frac{\partial Z}{\partial t} \, d\eta + \int_{S_p} g \frac{\partial \tilde{\phi}}{\partial t} Z \, d\eta \quad (2.40) \end{aligned}$$

Next, we eliminate the derivatives of SG between Eqs. (2.39) and (2.40), obtaining the differential equation

$$\begin{aligned} \frac{\partial \phi}{\partial t} + U \nabla_x \phi &= \frac{1}{SG} \int_{S_p} g \frac{\partial \tilde{\phi}}{\partial t} Z \, d\eta + \frac{1}{SG} \int_{S_p} (\tilde{\phi} - \phi) (U - \frac{\partial s}{\partial t}) \cdot (\nabla_x g) Z \, d\eta + \\ &+ \frac{1}{SG} \int_{S_p} (\tilde{\phi} - \phi) g \frac{\partial Z}{\partial t} \, d\eta \quad (2.41) \end{aligned}$$

The first integral on the right-hand side of Eq. (2.41) is by definition the surface average of $\partial \tilde{\phi} / \partial t$. The other two integrals are assumed to be small and neglected for interior ballistics problems. We notice that both integrals vanish if $\phi = \tilde{\phi}$ on the propellant surface, i.e., if the property ϕ is identical for all grains. Also, the term $U - \partial s / \partial t$ can be assumed small, e.g., if all grains have the same velocity U and do not burn, because $\partial s / \partial t$ is equal to the sum of the grain velocity and surface regression velocity. The term $\partial Z / \partial t$ is zero if the grains are not burning.

If we neglect the last two integrals in Eq. (2.41) and use Eq. (2.26) to define

$$\dot{\phi} = \frac{1}{SG} \int_{S_p} g \frac{\partial \tilde{\phi}}{\partial t} dS(\eta) \quad (2.42)$$

then the differential equation, Eq. (2.41) simplifies to

$$\frac{\partial \phi}{\partial t} + UV_x \phi = \langle \dot{\phi} \rangle, \quad (2.43)$$

where $\langle \dot{\phi} \rangle$ is a model for $\dot{\phi}$.

For the velocity U one chooses an average grain velocity, assuming that by this choice one of the neglected terms can be kept small.

2.3 Regions of Definition of Averages

In this section we describe regions of definition of the average functions. In principle, the averaging volume V can be of any shape and size. However, in order to preserve a cylindrical symmetry of the averaged quantities, the volume V , the weight function g , and the reference point x associated with the location of the volume, all must be chosen with certain symmetry properties. Instead of trying to formulate a general averaging volume with the desired properties, we give two examples of admissible averaging volumes.

The simplest example of an averaging volume is a sphere with the reference point x in its center and a weight function that depends only on the distance from its center. Let the diameter of the sphere be l .

Another example is an orthogonal circular cylinder with the reference point at its center and with an axis parallel to the axis of the gun tube. To be specific, we assume that the height of the cylinder is $2l/3$ if l is the diameter of the cylinder. In this example, the weight function depends on the radial as well as on the axial coordinates within the cylinder.

In both examples, the quantity l is equal to the largest diameter of the averaging volume. In general, we may assume a characteristic length l associated with any particular averaging volume. The size of the volume and, therefore, the size of l , is restricted by two requirements. First, the averaging volume must fit inside the gun barrel and, second, we want it to be larger than the largest grain in order to insure that gas is present within every averaging volume. Let D_p be the largest diameter of a grain and let D_{gun} be the inner diameter of the gun tube. Then in the two examples l must satisfy the conditions

$$(\bar{D}_p^*)_{\max} < l < (D_{\text{gun}})_{\min} \quad (2.44)$$

Similar restrictions one would obtain for the characteristic length of any averaging volume. We assume that \bar{D}_p^* and D_{gun} are such that the inequalities in Eq. (2.44) can be satisfied by a margin if l is properly chosen.

The position of the averaging volume inside the gun tube is restricted. If a constant averaging volume intersects a boundary, then the sum of the gas volume fraction α , as defined by Eq. (2.3), and of the corresponding particle volume fraction is not equal to one. Consequently, the definition of averages by Eqs. (2.2) through (2.5) cannot be used if a non-zero intersection occurs, and the location of the averaging volume is restricted to positions without intersections between the averaging volume and boundaries. (See also Section 4.6.) This means that the reference point x cannot be moved arbitrarily close to all boundaries. If the averaging volume is a sphere with the diameter l , then x is restricted to locations that are at least $l/2$ away from the breech, the walls, and projectile base. In the second example (cylinder), x may be located at points that are at least $l/2$ away from the tube walls and $l/3$ away from the breech and from the projectile base. Consequently, because of the finite size of the averaging volume, none of the averaged quantities are defined in the boundary regions. If the grain diameter \bar{D}_p^* is large, then the regions where the averaged quantities are not defined can be a significant part of the interior of the gun tube.

In the remaining regions, the porosity α and all averages pertaining to gas properties are everywhere defined by Eqs. (2.3) and (2.4), respectively.

Average properties of propellant grains are defined by Eq. (2.5). The definition provides a value for the average function only if $\alpha < 1$, i.e., if there are grains within the averaging volume. The limitation also holds for surface averaged quantities, defined by Eq. (2.25). The surface averaged quantities are grain properties and they are defined only if there are grains within the averaging volume.

The weighted number \bar{m}^* of grains in the averaging volume is defined by

$$\bar{m}^*(t, x) = VG (1-\alpha)/v_p^*(\bar{d}) \quad (2.45)$$

where \bar{d} is the average regression distance of the grains and $v_p^*(\bar{d})$ is the corresponding grain volume, given by a correlation function. (Particulars of this definition are discussed in Section 4.2). According to the definition, \bar{m}^* is indeterminate in regions without grains, because \bar{d} is not defined in those regions. We notice, however that $\bar{m}^* \rightarrow 0$ and $\nabla \bar{m}^* \rightarrow 0$ as x moves to a position where the averaging volume contains no grains. Therefore, we may define a continuation $\bar{m}^* \equiv 0$ in regions without grains. With this extension, \bar{m}^* is defined in all those regions where gas properties are defined, i.e., everywhere, except in boundary regions.

2.4 Averaging Weight Function

The averaging weight function $g(y)$ is defined inside the averaging volume V and on its boundary S_V . It has the following properties (see Sections 2.2.2, 2.2.3, and 2.2.4)

$$\left. \begin{aligned} g &> 0 \quad \text{in } V, \\ g &= 0 \quad \text{on } S_V, \\ \nabla g &\text{ continuous in } V \text{ with possible exemption of isolated} \\ &\text{singular points}, \\ \nabla g &= 0 \quad \text{on } S_V. \end{aligned} \right\} (2.46)$$

Next, we give examples of functions $g(y)$ that satisfy these conditions for the two examples of averaging volumes mentioned in the previous section. Let $y = \xi - x$, i.e., let the point of origin of the coordinate vector y be at the center of the averaging volume. (In both our examples the center coincides with the reference point x .)

If V is a sphere with the diameter l , then we define

$$g(y) = \frac{(2+n)(3+n)(4+n)}{6} \left(1 - \frac{|y|}{l/2}\right)^{1+n}, \quad \text{for } -\frac{l}{2} < y < \frac{l}{2} \quad (2.47)$$

with an $n > 0$. The weighted averaging volume VG is for this $g(y)$

$$VG = \int_V g \, dV = 4\pi \int_0^{l/2} g(y) y^2 dy = \frac{4}{3} \pi \left(\frac{l}{2}\right)^3. \quad (2.48)$$

As a second example we chose a cylinder with the diameter l and height $2l/3$. Let r and z be the radial and axial coordinates within the cylinder, with the point of origin at the center of the cylinder. Then we define

$$g(r, z) = \frac{1}{2} (2+n) (2+n) (3+n) \left(1 - \frac{|z|}{l/3}\right)^{1+n} \left(1 - \frac{|r|}{l/2}\right)^{1+n}. \quad (2.49)$$

The weighted averaging volume VG is for this choice of g

$$VG = \int_V g \, dV = 4\pi \int_0^{\ell/3} \int_0^{\ell/2} g(r,z) r \, dr \, dz = \frac{4}{3} \pi \left(\frac{\ell}{2}\right)^3, \quad (2.50)$$

i.e., equal to the volume $|V|$ of the cylinder itself.

In both examples, we have weight functions with a maximum at the center of the averaging volume. The functions are continuous but have gradient singularities. The weight function for the spherical averaging volume has a singular point at the center of the sphere. The second weight function has a singular gradient along the line $r = 0$ and on the plane $z = 0$. Therefore, if the flow includes phenomena that require surface averaging one should use a different weight function for the cylindrical averaging volume. (For volume averaging, piecewise continuity of Vg is sufficient.)

The following two weight functions have no singularities. They are chosen such that the weighted averaging volume is the same as before, i.e., equal to the volume of a sphere with diameter ℓ .

A weight function example for a sphere is

$$g(r) = \frac{\pi^2}{\pi^2 - 6} \left[\cos \left(\pi \frac{r}{\ell/2} \right) + 1 \right]. \quad (2.51)$$

A weight function for the cylindrical averaging volume is

$$g(r,z) = \frac{\pi^2}{\pi^2 - 4} \left[\cos \left(\pi \frac{r}{\ell/2} \right) + 1 \right] \left[\cos \left(\pi \frac{z}{\ell/3} \right) + 1 \right]. \quad (2.52)$$

Numerous other examples can be constructed, e.g., based on the functions

$$g(r) = \left(1 - \left(\frac{r}{\ell/2} \right)^{2m} \right)^{1+n} \quad (2.53)$$

or

$$g(r) = \left[\cos \left(\frac{\pi}{2} \frac{r}{\ell/2} \right) \right]^{1+n} \quad (2.54)$$

and corresponding for the dependence on z . Particularly, functions of the type (2.53) with large m and small positive n have properties that are desirable according to Section 4.2.1.

3. CONSERVATION EQUATIONS

The mathematical description of a two-phase flow field is composed of two sets of local conservation equations (one for each phase), a set of local constitutive relations for each phase, and interfacial or jump conditions which relate locally the two phases. As in other two-phase models of interior ballistics, all chemical reactions are excluded. Burning of the grains is represented by a transfer of mass, momentum, and energy from the solid phase to the gas phase. Furthermore, the effects of body forces on both phases are assumed to be negligible. By averaging the local conservation equations according to the definitions and formulas determined in Section 2, and by using the local interfacial conditions, we derive the coupled set of average two-phase equations. The details of this procedure are given in this section. The average equations in vector form are derived in three space dimensions and time. The governing equations for cylindrically symmetric flow in cylindrical coordinates are listed componentwise in Appendix A.

3.1 Local Equations

3.1.1 Local Conservation Equations. The flow field is assumed to be composed of two disjoint phases: gas and solid grains. The gas is assumed to be compressible, viscous and heat conducting. The local conservation equations for the gas are the Navier-Stokes equations (Tsien, 1958, pp. 3-16)

$$\frac{\partial \tilde{\rho}}{\partial t} + \nabla \cdot (\tilde{\rho} \tilde{\mathbf{u}}) = 0 \quad , \quad (3.1)$$

$$\frac{\partial (\tilde{\rho} \tilde{\mathbf{u}})}{\partial t} + \nabla \cdot (\tilde{\rho} \tilde{\mathbf{u}} \tilde{\mathbf{u}}) = - \tilde{\nabla} \tilde{p} + \nabla \cdot \tilde{\Pi} \quad , \quad (3.2)$$

$$\frac{\partial (\tilde{\rho} \tilde{e})}{\partial t} + \nabla \cdot (\tilde{\rho} \tilde{\mathbf{u}} \tilde{e}) = - \tilde{p} \nabla \cdot \tilde{\mathbf{u}} + \tilde{\Phi}_1 - \nabla \cdot \tilde{\mathbf{Q}} \quad , \quad (3.3)$$

where $\tilde{\rho}$, \tilde{e} , and $\tilde{\mathbf{u}}$ are the density, specific internal energy, and the velocity vector, respectively. The constitutive laws for the viscous stress tensor $\tilde{\Pi}$, the heat dissipation function $\tilde{\Phi}_1$, and the heat conduction vector $\tilde{\mathbf{Q}}$ are

$$\tilde{\Pi} = 2\tilde{\mu} \tilde{\mathbf{E}} + (\tilde{\lambda} - \frac{2}{3} \tilde{\mu}) \nabla \cdot \tilde{\mathbf{u}} \mathbf{I} \quad , \quad (3.4)$$

$$\tilde{\Phi}_1 = 2\tilde{\mu} \tilde{\mathbf{E}} : \tilde{\mathbf{E}} + (\tilde{\lambda} - \frac{2}{3} \tilde{\mu}) (\nabla \cdot \tilde{\mathbf{u}})^2 \quad , \quad (3.5)$$

$$\tilde{\mathbf{Q}} = - \tilde{\kappa} \nabla \tilde{T} \quad , \quad (3.6)$$

where

$$\tilde{E} = 0.5 [\nabla \tilde{u} + (\nabla \tilde{u})^T] , \quad (3.7)$$

and $\tilde{\mu}$, $\tilde{\lambda}$, $\tilde{\kappa}$ are the shear viscosity coefficient, the bulk viscosity coefficient and the heat conduction coefficient, respectively, that may depend on the local temperature \tilde{T} . The local pressure and temperature are given by equations of state of the form $\tilde{p} = \tilde{p}(\tilde{\rho}, \tilde{e})$ and $\tilde{T} = \tilde{T}(\tilde{\rho}, \tilde{e})$.

Each solid grain is assumed to be incompressible (the density of a grain $\tilde{\rho} = \text{constant}$) but deformable. The local conservation equations for the solid phase can be expressed in a form similar to those of Eqs. (3.1) and (3.2) (Prager, 1961):

$$\frac{\partial}{\partial t} (\tilde{\rho}) + \nabla \cdot (\tilde{\rho} \tilde{u}) = 0 , \quad (3.8)$$

$$\frac{\partial}{\partial t} (\tilde{\rho} \tilde{u}) + \nabla \cdot (\tilde{\rho} \tilde{u} \tilde{u}) = \nabla \cdot \tilde{\Pi} , \quad (3.9)$$

where \tilde{u} is the local velocity vector of the grain. For our purposes, the solid phase stress tensor $\tilde{\Pi}$ represents the total stress within the solid grain. A constitutive law for $\tilde{\Pi}$ could be based on Hooke's law. Although the local angular momentum of the grains could be significant, it is assumed that the average effect of the angular momentum is small and can be neglected. Consequently, the local conservation equation for the angular momentum of a grain is omitted.

3.1.2 Local Interfacial Conditions. The interfacial conditions relate the two disjoint phases. The interface between the gas and solid is considered a singular surface across which mass, momentum and energy is transferred. The conditions that are valid on the interface can be expressed as (Truesdell and Toupin, 1960):

$$n \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) = n \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) , \quad (3.10)$$

$$n \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{u} + n \cdot \tilde{p} - n \cdot \tilde{\Pi} = n \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{u} - n \cdot \tilde{\Pi} , \quad (3.11)$$

$$\begin{aligned} n \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \left[\tilde{e} + \frac{1}{2} \tilde{u} \cdot \tilde{u} \right] + \tilde{p} n \cdot \tilde{u} + n \cdot \tilde{Q} - n \cdot \tilde{\Pi} \cdot \tilde{u} \\ = n \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \left[\tilde{e} + \frac{1}{2} \tilde{u} \cdot \tilde{u} \right] + n \cdot \tilde{Q} - n \cdot \tilde{\Pi} \cdot \tilde{u} , \end{aligned} \quad (3.12)$$

where \tilde{u}_{sp} is the local interface velocity, n is a unit normal, and \tilde{Q} is the local heat conduction vector within the grain.

The local interface velocity \tilde{u}_{sp} is defined in terms of the local regression rate $\dot{\tilde{d}}$ of the grain surface

$$\tilde{u}_{sp}(t, \xi(n)) = \tilde{u}(t, \xi(n)) + n_{sp} \dot{\tilde{d}}(t, \xi(n)) \quad , \quad (3.13)$$

where $\dot{\tilde{d}} > 0$ and n_{sp} is the outward unit normal to the grain with respect to the gas.

3.2 Averaging of the Local Conservation Equations

3.2.1 Derivation of the Average Gas Continuity Equation and Porosity Equation. To derive the average gas phase continuity equation, we multiply Eq. (3.1) by $\beta(t, \xi)g(\xi-x)$, integrate over the averaging volume $V(x)$ and obtain

$$\begin{aligned} \int_{V(x)} \beta(t, \xi)g(\xi-x) \frac{\partial \tilde{\rho}}{\partial t}(t, \xi) dV(\xi) \\ + \int_{V(x)} \beta(t, \xi)g(\xi-x) \nabla_{\xi} \cdot [\tilde{\rho}(t, \xi)\tilde{u}(t, \xi)] dV(\xi) = 0 \quad . \end{aligned} \quad (3.14)$$

Using formulas (2.9) and (2.21) with respect to the first and second integrals of (3.14), respectively, we have

$$\frac{\partial}{\partial t} \int_V \beta g \tilde{\rho} dV + \nabla_x \cdot \int_V \beta g \tilde{\rho} \tilde{u} dV + \int_{S_p} g \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \cdot n_{sp} dS = 0 \quad . \quad (3.15)$$

By the definition of a volume averaged quantity (2.4) and the interfacial mass flux condition (3.10), Eq. (3.15) can be written as

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha(t, x)\rho(t, x)) + \nabla \cdot (\alpha(t, x) \boxed{\rho u}(t, x)) \\ + \frac{\rho}{VG} \int_{S_p} g(\tilde{u} - \tilde{u}_{sp}) \cdot n_{sp} dS = 0 \quad , \end{aligned} \quad (3.16)$$

because $\tilde{\rho} = \rho^* = \text{constant}$ and $VG = \text{constant}$. In Eq. (3.16) ρ is the average gas density and the quantity $\overline{\rho u}$ is the average of the gas momentum density ρu . We define the average gas velocity vector u as the ratio

$$u(t, x) \equiv \frac{\overline{\rho u}(t, x)}{\rho(t, x)} . \quad (3.17)$$

Using this definition of u , the local regression rate, defined by Eq. (3.13), and the definition of the average surface function (2.25), we can rewrite the average gas continuity equation (3.16) as

$$\frac{\partial}{\partial t} [\alpha(t, x) \rho(t, x)] + \nabla \cdot [\alpha(t, x) \rho(t, x) u(t, x)] = \rho^* \frac{SG(t, x)}{VG} \dot{d}(t, x) . \quad (3.18)$$

The derivation of the average solid phase continuity equation proceeds in a similar fashion to that of the average gas continuity equation. Multiplying Eq. (3.8) by $(1-\beta)g$, integrating over $V(x)$, invoking formulas (2.10) and (2.22), and using the definitions (2.5) of an average solid grain property, and (3.13) of the local regression rate, we have

$$\frac{\partial}{\partial t} (VG(1-\alpha)\rho^*) + \nabla \cdot (VG(1-\alpha) \overline{\rho u}^*) - \rho^* \int_{S_p} g \tilde{d} dS = 0 . \quad (3.19)$$

Using the surface average definition (2.25) and the fact that ρ^* is a constant, Eq. (3.19) can be written as

$$\frac{\partial}{\partial t} (1-\alpha) + \nabla \cdot [(1-\alpha)u] = - \frac{SG}{VG} \dot{d} . \quad (3.20)$$

Hence, for incompressible solid grains, the average continuity equation for the solid phase, Eq. (3.20), is the governing equation for the porosity α .

We notice that, if $\tilde{\rho}$ is constant or depends only on time, then the average grain velocity \tilde{u} is given directly by Eq. (2.5). The different definition of the average u by Eq. (3.17) via the average momentum density $\overline{\rho u}$ is advantageous when $\tilde{\rho}$ is not constant.

The average gas continuity equation, Eq. (3.18), is coupled to the solid phase by the source term $\rho^*(SG/VG)\dot{d}$. As expected, the amount of mass added to the gas phase is exactly the amount liberated from the solid phase. If the grains are not regressing (burning), the average regression rate \dot{d} and the source term are zero. The surface average SG and the surface average regression rate \dot{d} are two new unknowns. To restrict the number of unknowns, \dot{d}

is replaced by a correlation (denoted by $\langle \dot{d} \rangle$) which is obtained from experiments (see Section 4.7.7). To understand the error involved in such a substitution, we rewrite Eq. (3.18) as

$$\begin{aligned} \frac{\partial}{\partial t}[\rho] + \nabla \cdot [\rho u] = \rho \frac{SG}{VG} \langle \dot{d}(t, x) \rangle \\ + \rho \frac{SG}{VG} \left[\frac{1}{SG} \int_{S_p} g(\xi(n) - x) \tilde{\dot{d}}(t, \xi(n)) dS(n) - \langle \dot{d}(t, x) \rangle \right]. \end{aligned} \quad (3.21)$$

The bracketed term on the right-hand side of Eq. (3.21) is the error term and is equal to

$$\frac{1}{SG} \int_1 \tilde{\dot{d}}(t, \xi(\hat{n}_1)) \int_{sp_1} g(\xi(n) - x) dS(n) - \langle \dot{d}(t, x) \rangle \quad (3.22a)$$

by the mean value theorem for multiple integrals (Apostol, 1957) and where \hat{n}_1 is some point on sp_1 . From expression (3.22a), the following inequality can be derived:

$$\left| \frac{1}{SG} \int_{S_p} g \tilde{\dot{d}} dS(n) - \langle \dot{d} \rangle \right| < \max_i |\dot{d}(t, \xi(\hat{n}_1)) - \langle \dot{d}(t, x) \rangle|. \quad (3.22b)$$

Thus, a sufficient condition for the error to be small is that the fluctuations of the values of the local regression rate \dot{d} over each surface from the value of the correlation $\langle \dot{d} \rangle$ at point x are small. A common expression for $\langle \dot{d} \rangle$ is given by Eq. (4.100). If the error given by Eq. (3.22a) is not small, another correlation for $\langle \dot{d} \rangle$ must be used. In practice, the error is assumed small and Eqs. (3.18) and (3.20) are written with \dot{d} replaced by $\langle \dot{d} \rangle$. Furthermore, an additional formal error could be introduced by the modeling of SG. However, this is avoided by the definition of \dot{m}^* in terms of SG (see Section 4.7.8).

3.2.2 Derivation of the Average Gas and Solid Momentum Equations. The average gas momentum equation is derived by multiplying the local momentum equation, Eq. (3.2), by the function βg , by integrating over the averaging volume $V(x)$ and by applying formulas (2.9) and (2.21). The results of these operations can be written as

$$\begin{aligned}
& \frac{\partial}{\partial t} \int_V \beta g \tilde{\rho} \tilde{u} dV - \int_{S_p} g \tilde{\rho} n_{sp} \cdot \tilde{u}_{sp} dS + \nabla_x \cdot \int_V \beta g \tilde{\rho} \tilde{u} \tilde{u} dV \\
& + \int_{S_p} g n_{sp} \cdot \tilde{\rho} \tilde{u} \tilde{u} dS = - \nabla_x \cdot \int_V \beta g \tilde{p} dV + \nabla_x \cdot \int_V \beta g \tilde{\Pi} dV \quad (3.23) \\
& - \int_{S_p} g (n_{sp} \tilde{p} - n_{sp} \cdot \tilde{\Pi}) dS .
\end{aligned}$$

We use the definition of an average gas property (2.4) and the definition of u (3.17) in Eq. (3.23) to obtain

$$\begin{aligned}
& \frac{\partial}{\partial t} [\alpha(t, x) \rho(t, x) u(t, x)] + \nabla \cdot [\alpha(t, x) \boxed{\rho u u}(t, x)] \\
& = - \nabla \cdot \left\{ \frac{1}{VG} \int_V \beta g \tilde{p} dV \right\} + \nabla \cdot \left\{ \frac{1}{VG} \int_V \beta g \tilde{\Pi} dV \right\} \quad (3.24) \\
& - \frac{1}{VG} \int_{S_p} g \{ n_{sp} \tilde{p} - n_{sp} \cdot \tilde{\Pi} + n_{sp} \cdot \tilde{\rho} [\tilde{u} - \tilde{u}_{sp}] \tilde{u} \} dS .
\end{aligned}$$

The term $\boxed{\rho u u}(t, x)$ represents the average of the product $\tilde{\rho} \tilde{u} \tilde{u}$. Because the average quantities ρ and u are already defined, we can denote the fluctuations of the values of the local variables from the value of the average variables as

$$\tilde{\rho}'(t, \xi) = \tilde{\rho}(t, x) - \rho(t, \xi) ,$$

and

$$(3.25)$$

$$\tilde{u}'(t, \xi) = \tilde{u}(t, x) - u(t, \xi) .$$

If we substitute formulas (3.25) into the integral representation of $\alpha \boxed{\rho u u}$, we obtain

$$\frac{1}{VG} \int_V \beta g \tilde{\rho} \tilde{u} \tilde{u} dV = \alpha \rho u u + \frac{1}{VG} \int_V \beta g \tilde{\rho} \tilde{u}' \tilde{u}' dV . \quad (3.26)$$

The difference between the first term on the right-hand side of Eq. (3.26) and the left-hand side, involves a volume average of the product of velocity fluctuations. We define this difference as the turbulent stress tensor of the flow. Thus, turbulence in this report is defined as volume averaged fluctuations. The turbulent stress tensor Π_T models the quantity

$$-\frac{1}{\alpha} \frac{1}{VG} \int_V \beta g \tilde{\rho} \tilde{u} \tilde{u} \, dV = \rho u u - [\rho u u] \quad (3.27)$$

We shall not discuss particular turbulence models in this report. A model is proposed in Gibeling et al. (1980). If we write the integral representation of $[\rho u u]$ and apply the mean value theorem for multiple-integrals (Apostol, 1957), we can rewrite Eq. (3.27) when V_{gas} is a connected set as

$$[u(t, x)u(t, x) - \tilde{u}(t, \hat{\xi})\tilde{u}(t, \hat{\xi})] \rho(t, x) \quad , \quad (3.28)$$

where $\hat{\xi}$ lies in V_{gas} and is different for each component of the tensor $\tilde{u}\tilde{u}$. From Eq. (3.28), a good model of the turbulent stress tensor for compressible flows is one which models the significant differences in the dyad of the average velocities and the dyad of the local velocities componentwise. With respect to the errors generated by the model Π_T in Eq. (3.24), we want the errors in the vector

$$\nabla \cdot \{ \alpha \Pi_T - [\alpha \rho u u - \alpha [\rho u u]] \} \quad (3.29)$$

to be minimized by the model.

Substituting Eq. (3.27) into Eq. (3.24) and algebraically manipulating the result, we have

$$\begin{aligned} \frac{\partial}{\partial t} [\alpha \rho u] + \nabla \cdot [\alpha \rho u u] = & -\alpha \nabla p + \nabla \cdot (\alpha \Pi) + \nabla \cdot (\alpha \Pi_T) \\ & + \rho \frac{SG}{VG} \tilde{u} \langle \dot{d} \rangle - \left[\frac{1}{VG} \int_S g (\tilde{n}_{sp} \tilde{p} - \tilde{n}_{sp} \cdot \tilde{\Pi}) dS + p \nabla \alpha \right] \\ & - \left\{ \frac{1}{VG} \int_S \left[\tilde{n}_{sp} \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{u} - \tilde{n}_{sp} \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{u} \right] dS \right. \\ & + \left. \left[-\frac{1}{VG} \int_S \tilde{n}_{sp} \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{u} dS - \rho \frac{SG}{VG} \tilde{u} \langle \dot{d} \rangle \right] \right\} \\ & + \left\{ \nabla \cdot \left[\frac{1}{VG} \int_V \beta g (\rho u u - \tilde{\rho} \tilde{u} \tilde{u}) dV - \alpha \Pi_T \right] \right. \\ & - \left. \left[\nabla \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{p} dV - \alpha p \right] \right] \right. \\ & + \left. \left[\nabla \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{\Pi} dV - \alpha \Pi \right] \right] \right\} \quad (3.30) \end{aligned}$$

where p and Π are the constitutive models for the average pressure defined by $[\int_V \beta g p dV] / [\alpha \cdot VG]$ and the average viscous stress tensor defined by $[\int_V \beta g \Pi dV] / [\alpha \cdot VG]$, respectively. In general, it is simpler to model the average pressure and viscous stress tensor than to actually integrate the local constitutive laws. Each term in Eq. (3.30) which is enclosed by braces is an error term. We now shall discuss each error.

The errors in the models p , Π , Π_T , and those introduced by $\dot{u}^* \langle \dot{d} \rangle$ are represented by the last four terms on the right-hand side of Eq. (3.30). If V_{gas} is connected, the errors in the last two terms can be written as

$$\nabla_x \left[\frac{1}{VG} \int_V \beta g p dV - \alpha p \right] = \nabla \left\{ \alpha(t, x) [\tilde{p}(t, \hat{\xi}(x)) - p(t, x)] \right\} \quad (3.31)$$

and

$$\nabla_x \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{\Pi} dV - \alpha \Pi \right] = \nabla \cdot \left\{ \alpha(t, x) [\tilde{\Pi}(t, \hat{\xi}(x)) - \Pi(t, x)] \right\}, \quad (3.32)$$

where $\hat{\xi}(x)$ are the mean value points in $V_{gas}(t, x)$ which, in general, differ for p and for each component of the tensor $\tilde{\Pi}$. The models p and Π as well as the errors (3.31) and (3.32) are discussed in Sections 4.7.1 and 4.7.2, respectively. For the gas momentum equation, the best approximations for p and Π are the ones which minimize both, the differences in their values and their derivatives.

The error in the turbulence model is discussed previously in this section.

Using the definition of \tilde{d} (3.13) the second braced term in Eq. (3.30) can be written as

$$\dot{u}^* \frac{SG}{VG} \left\{ \sum_1 \frac{1}{SG} \int_{sp_1} \tilde{u}(t, \xi(n)) \tilde{d}(t, \xi(n)) dS(n) - \dot{u}(t, x) \langle \dot{d}(t, x) \rangle \right\}. \quad (3.33)$$

If both, \tilde{u} and \tilde{d} , are functions of t only, then expression (3.33) is zero and no error exists. When this is not the case, one can bound (3.33) using the mean value theorem for multiple integrals by

$$|\dot{u}^* \frac{SG}{VG}| \max_1 |\tilde{u}(t, \xi(\hat{n}_1)) \dot{d}(t, x) - \dot{u}(t, x) \langle \dot{d}(t, x) \rangle|, \quad (3.34)$$

where \hat{n}_1 is different on each surface sp_1 . Expression (3.34) can be bounded by

$$\frac{1}{\rho} \frac{SG}{VG} \left| \left\{ \dot{d}(t, x) \max_1 |\ddot{u}(t, \xi(\eta_1)) - \dot{u}(t, x)| + |\dot{u}(t, x)| |\langle \dot{d}(t, x) \rangle - \dot{d}(t, x)| \right\} \right| \quad (3.35)$$

Thus, the error in replacing $\frac{1}{SG} \int_{S_p} g \ddot{u} \dot{d} dS$ with $\dot{u} \langle \dot{d} \rangle$ consists of two parts. One error involves the approximation of \dot{d} with $\langle \dot{d} \rangle$ and is discussed in Section 3.2.1. The other term is small if the values of the local particle velocity at the grain surfaces are near that of the average particle velocity at x ; that is if the fluctuations are small. If both terms are not small, then a correlation of the fluctuations $(\ddot{u} \dot{d})'$ must be modeled and included in Eq. (3.30).

The term

$$\frac{1}{VG} \int_{S_p} [n_{sp} \cdot \ddot{\rho}(\ddot{u} - \ddot{u}_{sp}) \ddot{u} - n_{sp} \cdot \ddot{\rho}(\ddot{u} - \ddot{u}_{sp}) \ddot{u}] dS \quad (3.36)$$

can be rewritten using the mass flux jump Eq. (3.10) and regression rate definition (3.13) as

$$\frac{1}{\rho} \frac{1}{VG} \int_{S_p} (\ddot{u} - \ddot{u}_{sp}) \ddot{d} dS, \quad (3.37)$$

or using the momentum flux jump Eq. (3.11) as

$$\frac{1}{VG} \int_{S_p} [(n_{sp} \cdot \ddot{\Pi} - n_{sp} \ddot{p}) - n_{sp} \cdot \ddot{\Pi}] dS. \quad (3.38)$$

On the interface between the gas and the particles, we assume either that the normal stresses are equal (the integrand in Eq. (3.38) is zero), or equivalently, that the gas and particle velocities are equal (the difference in the integrand in Eq. (3.37) is zero). In the special case of no burning $\dot{d}=0$, the error is zero. When the above assumption is not true, the expression (3.36) must be modeled by a correlation.

From Eq. (2.21) with $\ddot{\phi} = 1$, we have the relationship

$$Va = - \frac{1}{VG} \int_{S_p} g n_{sp} dS. \quad (3.39)$$

Using the formula (3.39), we have the equality

$$\begin{aligned} & \frac{1}{VG} \int_{S_p} g[n_{sp} \tilde{p} - n_{sp} \cdot \tilde{\Pi}] dS + p \nabla \alpha \\ & = \frac{1}{VG} \int_{S_p} g[n_{sp} (\tilde{p} - p) - n_{sp} \cdot \tilde{\Pi}] dS . \end{aligned} \quad (3.40)$$

We define the surface integral on the right-hand side of Eq. (3.40) as the drag force. The drag force is modeled by the correlation D which is discussed in Section 4.7.5. The error incurred by this approximation is

$$\left\{ \int_{S_p} g[n_{sp} (\tilde{p} - p) - n_{sp} \cdot \tilde{\Pi}] dS - D(t, x) \right\} . \quad (3.41)$$

This definition is consistent with Ishii's (1975) development but is different from Gibeling et al. (1980) and Gough's (1974) which is defined in terms of the surface integral of the weighted fluctuation of the normal total gas stress tensor; $n_{sp} \cdot (\tilde{\Pi} - \tilde{\Pi}) - n_{sp} (p - \tilde{p})$. For the special case when the average viscous stress tensor is zero (the inviscid two-phase model), our definition and those of Gibeling et al. and Gough agree. We recognize the fact that Eq. (3.40) is a formal definition which may not be realized in an experimentally determined correlation. In such a case, the other effects would have to be included in Eq. (3.40).

The derivation of the average solid phase momentum equation parallels that for the average gas momentum equation. We multiply Eq. (3.9) by $(1-\beta)g$, integrate over the averaging volume V , use formulas (2.10) and (2.22) and the definition of the average of a solid grain property (2.5) to obtain

$$\begin{aligned} & \frac{\partial}{\partial t} [(1-\alpha(t, x)) \overline{\rho u}^{**}(t, x)] + \nabla \cdot [(1-\alpha(t, x)) \overline{\rho u u}^{***}(t, x)] \\ & = \nabla_x \cdot \left\{ \frac{1}{VG} \int_V (1-\beta) g \tilde{\Pi}(t, \xi) dV \right\} \\ & + \int_{S_p} g n_{sp} \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{u} dS - \int_{S_p} g n_{sp} \cdot \tilde{\Pi} dS . \end{aligned} \quad (3.42)$$

Because $\tilde{\rho}$ is a constant, $\overline{\rho u}^{**}(t, x) = \tilde{\rho} \tilde{u}(t, x)$ and $\overline{\rho u u}^{***}(t, x) = \tilde{\rho} \overline{u u}^{**}(t, x)$. By adding and subtracting $\nabla[(1-\alpha)p]$, by using Eq. (3.39), and replacing $n_{sp} \cdot \tilde{\Pi}$ on the surface with its equivalent via the momentum flux interfacial jump condition (3.11), we can rewrite Eq. (3.42) as

$$\begin{aligned}
\frac{\partial}{\partial t} [(1-\alpha) \rho \ddot{u}] + \nabla \cdot [(1-\alpha) \rho \ddot{uu}] &= - (1-\alpha) \nabla p \\
+ \nabla_x \cdot \left\{ \frac{1}{VG} \int_V (1-\beta) g \ddot{\Pi} dV + (1-\alpha) p I \right\} \\
+ \frac{1}{VG} \int_{S_p} g n_{sp} \cdot \ddot{\rho} (\ddot{u} - \ddot{u}_{sp}) \ddot{u} dS \\
+ \frac{1}{VG} \int_{S_p} g [n_{sp} \ddot{p} - n_{sp} p - n_{sp} \cdot \ddot{\Pi}] dS \\
+ \frac{1}{VG} \int_{S_p} g [n_{sp} \cdot \ddot{\rho} (\ddot{u} - \ddot{u}_{sp}) \ddot{u} - n_{sp} \cdot \ddot{\rho} (\ddot{u} - \ddot{u}_{sp}) \ddot{u}] dS ,
\end{aligned} \tag{3.43}$$

where I is the identity tensor. Eq. (3.43) can be rewritten as

$$\begin{aligned}
\frac{\partial}{\partial t} [(1-\alpha) \rho \ddot{u}] + \nabla \cdot [(1-\alpha) \rho \ddot{uu}] &= - (1-\alpha) \nabla p + \nabla \cdot [(1-\alpha) \ddot{\Pi}] \\
+ \nabla \cdot [(1-\alpha) \ddot{\Pi}_T] - \rho \frac{SG}{VG} \ddot{u} \langle \dot{d} \rangle + \frac{1}{VG} D \\
+ \left\{ \frac{1}{VG} \int_{S_p} g [n_{sp} \cdot \ddot{\rho} (\ddot{u} - \ddot{u}_{sp}) \ddot{u} - n_{sp} \cdot \ddot{\rho} (\ddot{u} - \ddot{u}_{sp}) \ddot{u}] dS \right. \\
- \left. \frac{1}{VG} \int_{S_p} g n_{sp} \cdot \ddot{\rho} (\ddot{u} - \ddot{u}_{sp}) \ddot{u} dS + \rho \frac{SG}{VG} \ddot{u} \langle \dot{d} \rangle \right\} \\
+ \{ \nabla \cdot [\rho (1-\alpha) (\ddot{uu} - \ddot{uu})] - (1-\alpha) \ddot{\Pi}_T \} \\
+ \{ \nabla \cdot \left[\left(\frac{1}{VG} \int_V (1-\beta) g \ddot{\Pi} dV + (1-\alpha) p I - (1-\alpha) \ddot{\Pi} \right) \right] \} \\
+ \left\{ \frac{1}{VG} \int_{S_p} g [n_{sp} (\ddot{p} - p) - n_{sp} \cdot \ddot{\Pi}] dS - \frac{1}{VG} D \right\} ,
\end{aligned} \tag{3.44}$$

where $\ddot{\Pi}$ is the constitutive model for the average stress tensor for the solid phase and represents

$$\frac{1}{1-\alpha} \frac{1}{VG} \int_V (1-\beta) g \ddot{\Pi} dV + p I , \tag{3.45}$$

and $\bar{\Pi}_T^*$ is the constitutive model for the average solid phase turbulent stress tensor. In analogy to $\bar{\Pi}_T$, $\bar{\Pi}_T^*$ models the dyad (see Eq. (3.27))

$$\bar{u}(t,x)\bar{u}(t,x) - \boxed{\bar{u}\bar{u}}(t,x) = -\frac{1}{1-\alpha} \frac{1}{VG} \int_V (1-\beta)g \tilde{u}^* \tilde{u}^* dV \quad (3.46)$$

We recall that by definition $\bar{\Pi}^*$ denotes the total stress tensor for the solid grain. The quantity defined by Eq. (3.45) is the difference between the average total stress in the solid phase (the integral of $(1-\beta)g\bar{\Pi}^*/VG$ over the averaging volume) and the stress caused by the average gas pressure $(-\pi I)$. The resulting stress is the stress caused by the grains themselves, for example, by the compactification of the propellant bed. Consequently, we call the expression (3.45), the average intergranular stress, and $\bar{\Pi}^*$ the average intergranular stress model. As with the average pressure, viscous stress tensor, and turbulent gas stress tensor, it is simpler to separately model the intergranular stress, the solid phase turbulent stress, and the drag. The errors incurred by these models are represented by the last three terms in Eq. (3.44).

The remaining error terms in Eq. (3.44) (those enclosed by braces) are the surface integral involving the velocity or stress jump, and the surface integral representing the source term. These terms are discussed in the derivation of the average gas momentum equation (see the analyses beginning near Eqs. (3.36) and (3.33), respectively).

3.2.3 Derivation of the Average Gas Internal Energy Equation. The average internal energy is needed to compute certain quantities, e.g., the pressure and temperature via the equations of state for the average quantities. The average internal energy can be obtained in either of two ways. First, by adding the local internal energy equation to the equation for the local kinetic energy, an equation for the local total energy can be written. Following a similar procedure to those given in Sections (3.2.1) and (3.2.2), we then can derive an average total energy equation. Finally, the average internal energy value is obtained as the difference between the average total energy, and the average kinetic energy determined from the average velocities. The second way is to average the local internal energy equation, Eq. (3.3), directly. The former procedure is the most common. However, we use the latter approach because several terms which must be assumed small or modeled by additional correlations can be avoided, and the terms, which must be modeled, have simpler physical interpretations, and, therefore, are easier to model. An example of a term that can be eliminated by the second method but is present in the first is

$$\int_V \beta(t, \xi) g(\xi - x) [\tilde{\rho}(t, \xi) \tilde{u}(t, \xi) \cdot \tilde{u}(t, \xi) - \rho(t, x) u(t, x) \cdot u(t, x)] dV \quad (3.47)$$

$$= \int_V \beta g \tilde{\rho} \tilde{u} \cdot \tilde{u} dV .$$

The non-negative expression (3.47) represents the average difference between the local kinetic energy from the dot product of the average velocity. An example of a term that can be modeled more easily in the average internal energy equation is the dissipation term. In the average internal energy equation, the term ϕ represents the average conversion of viscous work by the fluid into heat only. Whereas in the average total energy equation, the term $\nabla \cdot (\Pi \cdot u)$ models the average conversion of viscous work of the fluid into two quantities, heat and kinetic energy.

The average internal energy equation is derived in a similar fashion as the average gas continuity equation and gas momentum equations. We multiply Eq. (3.3) by βg , integrate over the averaging volume $V(x)$ and use formulas (2.9) and (2.21) to obtain

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \beta g \tilde{\rho} e dV + \nabla \cdot \int_V \beta g \tilde{\rho} u e dV &= - \int_{S_p} g n_{sp} \cdot \tilde{\rho} (\tilde{u} - \tilde{u}_{sp}) \tilde{e} dS \\ &- \int_V \beta g \tilde{p} \nabla \cdot \tilde{u} dV + \int_V \beta g \tilde{\phi}_1 dV - \nabla \cdot \int_V \beta g \tilde{Q} dV \\ &- \int_{S_p} g \tilde{Q} \cdot n_{sp} dS . \end{aligned} \quad (3.48)$$

We define the average specific internal energy e similar to the average gas velocity, that is, as the quotient of the average internal energy density $\overline{\rho e}$ and the average mass density ρ :

$$e = \frac{\overline{\rho e}(t, x)}{\rho(t, x)} . \quad (3.49)$$

Using Eqs. (3.13) and (3.49), Eq. (3.48) can be written, after some manipulation, as

AD-A118 920

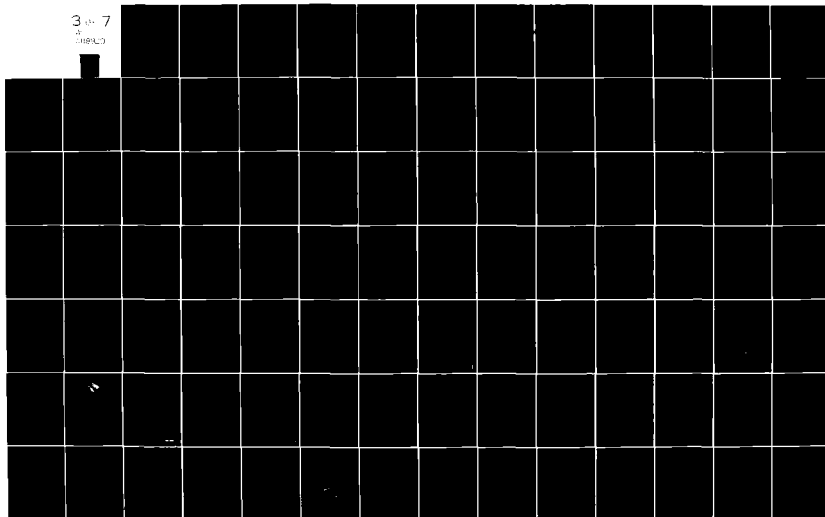
ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82
ARO-82-3

F/O 12/1

NL

UNCLASSIFIED

3-7
A
CIRCUIT



$$\begin{aligned}
\frac{\partial}{\partial t} (\alpha \rho e) + \nabla \cdot (\alpha \rho e \mathbf{u}) = & - \alpha \rho \nabla \cdot \mathbf{u} + \alpha \phi_L + \alpha \phi_T - \nabla \cdot (\alpha Q) \\
& + \rho \frac{SG}{VG} \hat{e} \langle \dot{d} \rangle - \frac{1}{VG} \int_{S_p} \mathbf{g} \tilde{Q} \cdot \mathbf{n}_{sp} dS \\
& + \nabla \cdot [\alpha \rho e \mathbf{u} - \alpha \boxed{\rho e \mathbf{u}}] + \left[\frac{1}{VG} \int_V \beta g \tilde{\theta}_1 dV - \alpha \phi_L - \alpha \phi_T \right] \quad (3.50) \\
& + \nabla \cdot [\alpha Q - \frac{1}{VG} \int_V \beta g \tilde{Q} dV] + \left[\alpha \rho \nabla \cdot \mathbf{u} - \frac{1}{VG} \int_V \beta g \tilde{\rho} \nabla \cdot \tilde{\mathbf{u}} dV \right] \\
& - \left[\rho \frac{SG}{VG} \hat{e} \langle \dot{d} \rangle - \frac{1}{VG} \int_{S_p} \tilde{g} \tilde{e} dS \right] ,
\end{aligned}$$

where ϕ_L , ϕ_T , and Q are the constitutive models for the average dissipation function, turbulent dissipation function, and the average heat conduction, respectively. The average energy release by the propellant during burning is denoted by $\dot{e}(t, \mathbf{x})$. The term $\alpha \rho e \mathbf{u} - \alpha \boxed{\rho e \mathbf{u}} = - (1/VG) \int_V \beta g \tilde{\rho} \tilde{e} \tilde{\mathbf{u}} dV$, in analogy to Eq. (3.26). This term is zero if either $\tilde{e} \equiv 0$, or $\tilde{\mathbf{u}} \equiv 0$, i.e., if e or u is a function of time only. However, in turbulent flows this term can be significant. A model of the term as the gradient of the energy variable is given by Cebeci and Smith (1974). In interior ballistics, the term is probably large, because for moving and burning grains the extrema of \tilde{e} and $\tilde{\mathbf{u}}$ are likely to correlate. We denote the model of this term by Q_T . The term $\int_{S_p} (\beta g \tilde{Q} \cdot \mathbf{n}_{sp} / SG) dS$ represents the average heat flux into the particle from the gas and is modeled by the correlation $\langle \dot{e} \rangle$. The models for ϕ_L and ϕ_T , Q and Q_T , and e and $\langle \dot{e} \rangle$ are discussed in Sections 4.7.3, 4.7.4, and 4.7.8, respectively.

We now can rewrite Eq. (3.50) as

$$\begin{aligned}
\frac{\partial}{\partial t} (\alpha \rho e) + \nabla \cdot (\alpha \rho e u) = & -\alpha p \nabla \cdot u + \alpha \dot{\phi}_L + \alpha \dot{\phi}_T - \nabla \cdot (\alpha Q) - \nabla \cdot (\alpha Q_T) \\
& + \frac{1}{\rho} \frac{SG}{VG} \hat{e} \cdot \langle \dot{d} \rangle - \frac{SG}{VG} \langle \dot{e} \rangle \\
& + \left\{ \frac{1}{VG} \int_V \beta g [\tilde{\phi}_1(t, \xi) - \phi_L(t, x) - \phi_T(t, x)] dV \right. \\
& - \left\{ \nabla \cdot \frac{1}{VG} \int_V \beta g [\tilde{Q}(t, \xi) - Q(t, x)] dV \right\} - \left\{ \nabla \cdot [(\alpha \boxed{\rho e u}) - \alpha \rho e u] - \alpha Q_T \right\} \\
& - \left\{ \frac{1}{VG} \int_{S_p} g [\tilde{Q}(t, \xi) \cdot n_{sp} - \langle \dot{e} \rangle] dS \right\} \\
& + \left\{ \frac{1}{VG} \int_V \beta g [p(t, x) \nabla \cdot u(t, x) - \tilde{p}(t, \xi) \nabla \cdot \tilde{u}(t, \xi)] dV \right\} \\
& - \left\{ \frac{1}{\rho} \frac{SG}{VG} [\hat{e} \cdot \langle \dot{d} \rangle - \frac{1}{VG} \int_{S_p} \tilde{g} \tilde{e} \tilde{d} dS] \right\} ,
\end{aligned} \tag{3.51}$$

where the terms enclosed by braces are error-type terms.

The first four error terms depend on a model and are discussed in the appropriate model section (see Section 4). The remaining two terms can be written by following similar analyses to those in the average gas momentum equation derivation as

$$\begin{aligned}
\frac{1}{VG} \int_V \beta g [p \nabla \cdot u - \tilde{p} \nabla \cdot \tilde{u}] dV = & \alpha(t, x) p(t, x) [\nabla \cdot u(t, x) - \nabla \cdot \tilde{u}(t, \hat{\xi}(x))] \\
& + \alpha(t, x) \nabla \cdot \tilde{u}(t, \hat{\xi}(x)) [p(t, x) - \tilde{p}(t, \hat{\xi}(x))]
\end{aligned} \tag{3.52}$$

and

$$\left| \rho^* \frac{SG}{VG} [\hat{e} \langle \dot{d} \rangle - \frac{1}{SG} \int_{S_p} g \tilde{e} \tilde{d} dS] \right| \quad (3.53)$$

$$< \left| \rho^* \frac{SG}{VG} \right| \{ \hat{e} | \langle \dot{d} \rangle - \dot{d} | + \dot{d} \max_1 |\hat{e}(t, x) - \tilde{e}(t, \xi(\eta_1))| \}$$

where $\xi(x)$ is a point in V_{gas} (V_{gas} is assumed connected) and η_1 is a point on the surface S_{p1} .

The error represented by Eq. (3.52) consists of two parts: the error made by using the divergence of the average velocity for the divergence of the local velocity, and the error made by using the average pressure correlation for the local pressure. If both \tilde{p} and $\tilde{V} \cdot \tilde{u}$ were functions of time only, the error would be zero. If the term is not negligible, then a correlation that models the average fluctuations of $\tilde{p} \tilde{V} \cdot \tilde{u}$ from $p \tilde{V} \cdot u$ must be included. Most often the term is neglected, but a model may be necessary in some turbulent flows. The error generated by replacing the surface integral of $g \tilde{e} \tilde{d} / SG$ with the product of correlations $e \langle \dot{d} \rangle$, Eq. (3.5.3), also consists of two parts. The first involves the approximation of \dot{d} by $\langle \dot{d} \rangle$ which is discussed in Section 3.2.1. The second is small if the fluctuations are small of the local internal energy from the specific internal energy of the gas at flame temperature, \hat{e} . In practice, both errors are assumed small. If not, a correlation which models the fluctuation of $\tilde{e} \dot{d}$ from $\hat{e} \langle \dot{d} \rangle$ over the surface of the grains must be included.

3.2.4 Derivation of the Surface Average Equations. On the surface of the particles, the average normal regression distance $\bar{\dot{d}}$ and the average surface temperature \bar{T} can be defined according to the definition (2.25), where \dot{d} and T denote the local values, respectively. According to Section 2.2.5, the variables $\bar{\dot{d}}$ and \bar{T} satisfy the differential equation (2.41) so that the average regression distance equation is

$$\begin{aligned} \frac{\partial \bar{\dot{d}}}{\partial t} + \bar{u} \cdot \nabla \bar{\dot{d}} = & \langle \dot{d} \rangle + \left\{ \frac{1}{SG} \int_{S_p} g \left(\frac{d\tilde{d}}{dt} - \langle \dot{d} \rangle \right) dS \right\} \\ & + \left\{ \frac{1}{SG} \int_{S_p} (\tilde{\dot{d}} - \bar{\dot{d}}) (\bar{u} - \tilde{u}_{sp}) \cdot \nabla_x g dS \right\} \\ & + \left\{ \frac{1}{SG} \int_{S_p} (\tilde{\dot{d}} - \bar{\dot{d}}) g \frac{\partial Z}{\partial t} d\eta \right\} , \end{aligned} \quad (3.54)$$

and the average surface temperature equation is

$$\begin{aligned} \frac{\partial \bar{T}}{\partial t} + \bar{u} \cdot \nabla \bar{T} = \langle \dot{T} \rangle + \left\{ \frac{1}{SG} \int_{S_p} g \left(\frac{dT}{dt} - \langle \dot{T} \rangle \right) dS \right\} \\ + \left\{ \frac{1}{SG} \int_{S_p} (\tilde{T} - \bar{T}) (\bar{u} - \tilde{u}_{sp}) \cdot \nabla_x g dS \right\} \\ + \left\{ \frac{1}{SG} \int_{S_p} (\tilde{T} - \bar{T}) g \frac{\partial Z}{\partial t} d\eta \right\} , \end{aligned} \quad (3.55)$$

where $\tilde{u}_{sp} = \frac{\partial g}{\partial t}$, $\langle \dot{T} \rangle$ is the correlation for the regression rate, and $\langle \dot{T} \rangle$ is the correlation for the rate of change of grain surface temperature.

The last three terms in each of the Eqs. (3.54) and (3.55) are error type terms. The first error terms in Eqs. (3.54) and (3.55) are the surface averages of the fluctuations between the local values and its corresponding correlation values of the regression rate and surface temperature, respectively. The regression rate term is discussed in Section 3.2.1 and similar error estimates and comments can be made concerning the surface temperature term. The remaining error terms involve fluctuations from formally defined averages. The last terms in Eqs. (3.54) and (3.55) involve fluctuations of \tilde{d} and \tilde{T} from their average values, respectively. Because the integrands of these surface integrals include other terms, these integrals are not surface averages of fluctuations, and, thus, are not necessarily zero. The other set of error terms include the product of the fluctuations of the local interface velocity from the volume average particle velocity \bar{u} with the fluctuations of \tilde{T} and \tilde{d} from their average values. As before, the integrals involving these products are not surface average integrals. If the fluctuations are small over the surface of all the particles, then the terms can be neglected. Such cases occur when the regression distance and/or the surface temperature of all the grains are equal. If these surface integrals represent significant contributions to the rate of change of the variables, correlations for them must be formulated and included in the governing equations (3.54) and (3.55).

3.3 Summary and Discussion of the Conservation Equations Without Error Terms

In this section, we will list and discuss the equations derived in Section 3.2 without error terms. We are aware that some of the neglected terms may be significant in some flows. In such cases, it (they) can be appended to the governing equation(s) and modeled. A good way to decide whether a term should be neglected or included in a set of equations is to compare the accurate solution of the equations with data from well-defined, carefully done experiments. Furthermore, we realize that some of the constitutive laws and correlations quite possibly can be coupled to each other

and terms in the governing equations could be grouped differently. Thus, the formal and physical meaning of some of the constitutive laws and correlations can change. Therefore, the form of the equations, correlations, and constitutive laws for interior ballistic applications listed in this report should not be considered as final.

The porosity equation (3.20) (the average solid phase continuity equation) can be written as

$$\frac{\partial}{\partial t} (1-\alpha) + \nabla \cdot [(1-\alpha)\mathbf{u}^*] = -\Gamma_1, \quad (3.56)$$

where the source term is given by

$$\Gamma_1 = \frac{SG}{VG} \langle \dot{d} \rangle. \quad (3.57)$$

The average solid phase momentum equation (3.44) expresses the conservation of the solid phase momentum density, and is

$$\begin{aligned} \frac{\partial}{\partial t} [(1-\alpha)\rho u^{**}] + \nabla \cdot [(1-\alpha)\rho u u^{***}] = & - (1-\alpha)\nabla p + (1-\alpha)\rho^* A_{\text{stress}} \\ & + (1-\alpha)\rho A_{\text{drag}} - \rho u \Gamma_1^{**}, \end{aligned} \quad (3.58)$$

where

$$(1-\alpha)\rho^* A_{\text{stress}} = \nabla \cdot [(1-\alpha)\Pi^* + (1-\alpha)\Pi_T^*], \quad (3.59)$$

and

$$(1-\alpha)\rho A_{\text{drag}} = \frac{1}{VG} D. \quad (3.60)$$

The average gas phase continuity equation (3.21) is

$$\frac{\partial}{\partial t} (\alpha\rho) + \nabla \cdot (\alpha\rho\mathbf{u}) = \rho^* \Gamma_1. \quad (3.61)$$

The average gas phase momentum equation (3.30) expresses the conservation of the momentum density and, with the definition of drag (3.40), can be written as

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha \rho u) + \nabla \cdot (\alpha \rho u u) = & - \alpha \nabla p + \alpha \rho A_{\text{visc}} + \alpha \rho A_{\text{turb}} \\ & + \rho u \Gamma_1^{**} - (1-\alpha) \rho A_{\text{drag}} \end{aligned} \quad (3.62)$$

where

$$\alpha \rho A_{\text{visc}} = \nabla \cdot (\alpha \Pi) \quad , \quad (3.63)$$

and

$$\alpha \rho A_{\text{turb}} = \nabla \cdot (\alpha \Pi_T) \quad . \quad (3.64)$$

The average gas phase energy equation (3.51) expresses the conservation of the gas phase internal energy density, and is

$$\frac{\partial}{\partial t} (\alpha \rho e) + \nabla \cdot (\alpha \rho e u) = - \alpha p \nabla \cdot u + \alpha \phi_1 + \alpha \nabla_1 + \rho e \Gamma_1^{**} \quad , \quad (3.65)$$

where

$$\phi_1 = \phi_L + \phi_T \quad , \quad (3.66)$$

and

$$\alpha \nabla_1 = - \nabla \cdot (\alpha Q) - \frac{SG}{VG} \langle \dot{e} \rangle - \nabla \cdot (\alpha Q_T) \quad . \quad (3.67)$$

The term ϕ_1 contains all the models for the heat dissipation functions and the term ∇_1 contains those for the heat conduction within the gas and to the particles, and the turbulent heat flux.

The governing equations for the surface average regression rate (3.54) and for the surface average surface temperature (3.55) are

$$\frac{\partial \bar{d}}{\partial t} + \bar{u} \cdot \nabla \bar{d} = \langle \dot{d} \rangle, \quad (3.68)$$

and

$$\frac{\partial \bar{T}}{\partial t} + \bar{u} \cdot \nabla \bar{T} = \langle \dot{T} \rangle. \quad (3.69)$$

Because the left-hand sides of these equations represent material derivatives, one can interpret the equations as state equations for the surface material.

The source term is modeled by $\frac{SG}{VG} \langle \dot{d} \rangle$ which appears in every volume averaged equation. Recalling the definition of the source term

$$\frac{SG}{VG} \dot{d}(t, x) = \frac{1}{VG} \int_{S_p} \tilde{\dot{d}} dS, \quad \tilde{\dot{d}} \geq 0, \quad (3.70)$$

we see that the model must be zero when no particles within the averaging volume at point (t, x) is burning (regression rate \dot{d} is zero). When no particles exist within the averaging volume we want the value of the source term to be zero. This is reasonable because for the case of uniformly regressing particles, the integral in Eq. (3.70) approaches zero as the porosity approaches one. Furthermore, the value of the model must be always non-negative. Comparing Eq. (3.61) and ρ times Eq. (3.56), we see that value of the average mass flux per volume to the gas phase is exactly that being taken away from the solid phase within the averaging volume. The average balance can also be seen in the momentum equations and involves the momentum flux model $\rho u \Gamma_1$. The drag force per volume, D/VG , is also balanced on the average in the momentum equations (3.58) and (3.62). We note that the model for the drag force D should be zero when no particles exist ($\alpha=1$) in the flow because then the drag force Eq. (3.40) is zero (S_p has zero surface area). Appropriate types of average stress tensors are also included in the average momentum equations. For the gas phase, Eq. (3.62), the average viscous stress tensor $\bar{\Pi}$ and average turbulent stress tensor $\bar{\Pi}_T$ are weighted with respect to the average volume of gas present. For the solid phase, Eq. (3.58), the average intergranular stress tensor $\bar{\Pi}$ and average solid phase turbulent stress tensor $\bar{\Pi}_T$ are weighted with respect to the average volume of solid phase present and are grouped together. The internal energy of the gas phase, Eq. (3.65), is augmented by the source term $\rho e \Gamma_1$. The appropriately weighted heat dissipation functions ϕ_L and ϕ_T (the contribution from turbulence) are grouped together. The average work done by the gas pressure is denoted by $-pVu$ and is

weighted by the porosity. The average heat flux between the gas and the solid is represented by $\frac{SG}{VG} \langle \dot{e} \rangle$. The correlation $\langle \dot{e} \rangle$ should be positive when the temperature of the gas is higher than that of the solid, negative in the opposite case and zero when the temperatures are the same or when no particles exist in the averaging volume. The average heat conduction in the gas is modeled by $V \cdot (\alpha Q)$. The turbulent heat flux vector is modeled similarly by $V \cdot (\alpha Q_T)$. The last three terms are grouped in one term V_1 . The surface averaged equations for the average regression distance (3.68) and surface temperature (3.69) have non-negative valued right-hand sides represented by the correlations $\langle \dot{d} \rangle$ and $\langle \dot{T} \rangle$, respectively.

The limiting case of no particles within a region is of particular interest in interior ballistic applications because such regions do exist inside a gun tube. The other limiting case of no gas does not exist in our applications and, thus, is of no practical interest. In the case of no particles ($\alpha=1$), the set of conservation equations greatly simplify. The source terms are zero and the drag and interface heat transfer terms are also zero. However, it is important to notice that, first, the gas phase equations do not reduce to the local equations (3.1) through (3.3). The simplified set ($\alpha=1$) differs in form from the local equations because it includes the turbulence terms, that is, $V \cdot (\Pi_T)$, $\alpha \Phi_T$, and $V \cdot (Q_T)$. This fact reminds us that the resulting set of equations is still a set of average equations for a finite averaging volume V . Secondly, even if the averages of all the products of fluctuations were zero (no turbulence), then the set of equations for the gas flow would have the same form as the local equations, but the solutions would not be the same in general. This is so, because the quantities ρ , u , and e are averaged, and their initial and boundary conditions are not the same as the initial and boundary conditions for $\bar{\rho}$, \bar{u} , and \bar{e} in general. Thirdly, if we let the averaging volume go to zero in the simplified set (with $\alpha=1$), the turbulence terms would be zero because the fluctuations are averaged over the averaging volume which has zero volume. In this case ($\alpha=1$ and $V(x) \rightarrow 0$) the averaged equations reduce in form to the local equations and the initial and boundary conditions should reduce to the local conditions. Thus, the solutions of the two sets would be identical. Fourthly, in the case of $\alpha=1$, the equations for \dot{d} and \dot{T} , Eqs. (3.68) and (3.69), are homogenous ($\langle \dot{d} \rangle = \langle \dot{T} \rangle = 0$) but a value of \dot{d} and \dot{T} can be computed from these equations if \bar{u} is defined. Although these values would be physically meaningless, they allow the solution to be computed numerically everywhere without tracing the internal boundaries of gas and mixture. Because these internal boundaries cannot be predicted ahead of time in a two-dimensional flow field, this provides a distinct numerical advantage. Fifthly, the average solid phase momentum equation is identically satisfied when $\alpha=1$. Thus, the components of the vector \bar{u} cannot be determined from Eq. (3.58), and the numerical advantages discussed with respect to \dot{d} and \dot{T} are lost. In fact, when an implicit numerical algorithm is used to solve Eqs. (3.56) through (3.69) directly for the variables ρ , α , u , \bar{u} , e , \dot{d} , and \dot{T} , it can be shown that the matrix equation which must be solved for a new time level of values is singular (the rank of the matrix is deficient) when $\alpha=1$. To avoid this situation, we can algebraically manipulate the porosity and

solid phase momentum equations into a non-conservative form where $\partial \mathbf{u}^*/\partial t$ has a coefficient one. Then the components of \mathbf{u}^* can be defined everywhere. Another advantage of solving for the values of \mathbf{u}^* , \mathbf{d}^* , and \mathbf{T}^* directly from their governing partial differential equations when $\alpha=1$ is that their values should be continuous at $\alpha=1$ if the equations approach a non-singular form at $\alpha=1$.

In Section 4.1, 4.2, and 4.3, we discuss better forms of the partial differential equations and another choice of dependent variables for numerical treatment.

4. GOVERNING EQUATIONS

4.1 Basic System of Governing Equations

A system of conservation equations for average flow properties was derived in Section 3. One obtains an equivalent system of differential equations by solving the conservation equations (3.56) through (3.69) for the time derivatives of the dependent variables. Let the ensuing system be called governing equations of the flow. It consists of the following set of equations

$$\left. \begin{aligned} \frac{\partial \rho}{\partial t} &= -\nabla \cdot (\rho \mathbf{u}) - \frac{\rho}{\alpha} [(1-\alpha)\nabla \cdot \mathbf{u}^* - (\mathbf{u} - \mathbf{u}^*) \cdot \nabla (1-\alpha)] + \left(\frac{\rho^* - \rho}{\alpha}\right) \Gamma_2, \\ \frac{\partial e}{\partial t} &= -\mathbf{u} \cdot \nabla e - \frac{p}{\rho} \nabla \cdot \mathbf{u} + \left(\frac{e^* - e}{\alpha}\right) \frac{\rho}{\rho} \Gamma_2 + \frac{1}{\rho} (\Phi_1 + \Psi_1), \\ \frac{\partial \mathbf{u}}{\partial t} &= -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p - \frac{1}{\alpha} (\mathbf{u} - \mathbf{u}^*) \frac{\rho}{\rho} \Gamma_2 - \frac{1-\alpha}{\alpha} A_{\text{drag}} + A_{\text{visc}} + A_{\text{turb}}, \\ \frac{\partial \mathbf{u}^*}{\partial t} &= -(\mathbf{u}^* \cdot \nabla) \mathbf{u}^* - \frac{1}{\rho^*} \nabla p + \frac{\rho}{\rho^*} A_{\text{drag}} + A_{\text{stress}}, \\ \frac{\partial \alpha}{\partial t} &= \nabla \cdot ((1-\alpha)\mathbf{u}^*) + \Gamma_2, \\ \frac{\partial \mathbf{d}^*}{\partial t} &= -\mathbf{u}^* \cdot \nabla \mathbf{d}^* + \langle \dot{\mathbf{d}} \rangle, \\ \frac{\partial \mathbf{T}^*}{\partial t} &= -\mathbf{u}^* \cdot \nabla \mathbf{T}^* + \langle \dot{\mathbf{T}} \rangle. \end{aligned} \right\} (4.1)$$

The system is closed by a number of correlation models that will be discussed in detail in Section 4.7. Presently, we merely give a short exposition of the corresponding terms in Eq. (4.1). The listed arguments of the correlation functions are only representative, indicating the most obvious dependences.

The actual models may depend on fewer or on more arguments. Also, all models depend implicitly or explicitly on the averaging volume and on the averaging weight function.

The equations of state enter the system in form of a relation for the pressure, viz.,

$$p = p(\rho, e) , \quad (\text{Pa}) \quad . \quad (4.2)$$

The mass source due to the phase change by combustion is represented by

$$\dot{r}_2 = (1 - \alpha) \frac{s_p^*(\dot{d})}{v_p^*(\dot{d})} \langle \dot{d} \rangle , \quad (1/s) , \quad (4.3)$$

where we define $\frac{SG}{VG} = (1-\alpha)s_p^*(\dot{d})/v_p^*(\dot{d})$, and $v_p^*(\dot{d})$ and $s_p^*(\dot{d})$ are the volume and surface correlations, respectively, for propellant grains with the regression distance \dot{d} . The quantity $\langle \dot{d} \rangle$ represents the regression rate correlation. Generally it is a function of the type

$$\langle \dot{d} \rangle = \langle \dot{d} \rangle (p, |u^* - u|, \partial p / \partial t) , \quad (\text{m/s}) \quad . \quad (4.4)$$

The heat dissipation is modeled by the function

$$\dot{\phi}_1 = \dot{\phi}_1(u, T, \alpha, u^*, \dot{d}) , \quad (\text{W/m}^3) , \quad (4.5)$$

where $T(\rho, e)$ is provided by the equation of state correlation. The heat conduction is represented by the function

$$\dot{\gamma}_1 = \dot{\gamma}_1(T, \nabla T, \nabla \cdot \nabla T, \langle \dot{T} \rangle) , \quad (\text{W/m}^3) \quad . \quad (4.6)$$

The last argument of $\dot{\gamma}_1$ in Eq. (4.6) is the rate of change of the grain surface temperature, which may be modeled, e.g., by

$$\langle \dot{T} \rangle = \langle \dot{T} \rangle (\dot{T}, T, |u - u^*|) , \quad (\text{K/s}) \quad . \quad (4.7)$$

The term A_{drag} represents the acceleration due to the drag between gas and particles

$$A_{\text{drag}} = A_{\text{drag}} ((\dot{u} - u), \dot{d}, T), \quad (\text{m/s}^2) \quad (4.8)$$

The velocity governing equations contain three more acceleration terms. They are, the acceleration by the laminar viscosity

$$A_{\text{visc}} = A_{\text{visc}} (T, \nabla u, \nabla \cdot \nabla u, \alpha), \quad (\text{m/s}^2), \quad (4.9)$$

the acceleration due to turbulence

$$A_{\text{turb}} = A_{\text{turb}} (T, \nabla u, \nabla \cdot \nabla u, \dots), \quad (\text{m/s}^2), \quad (4.10)$$

and the acceleration due to intergranular stress and solid phase turbulence

$$A_{\text{stress}} = A_{\text{stress}} (\alpha, \dot{d}, \nabla u, \dots), \quad (\text{m/s}^2). \quad (4.11)$$

The system of governing equations, Eqs. (4.1), is for numerical solution more advantageous than the system of conservation equations (3.56) through (3.69) because none of the Eqs. (4.1) become identically satisfied as $\alpha \rightarrow 1$. This permits one to carry out the calculations throughout the interior of the gun tube without tracking the boundaries of regions with $\alpha = 1$.

We can further improve the equation system by selecting a new set of dependent variables. The choice of the new variables and the corresponding new system of governing equations are described in Sections 4.2 and 4.3, respectively.

4.2 Choice of Dependent Variables

4.2.1 Particle Number Function. If the source term Γ_2 is computed using Eq. (4.3), then one can expect numerical difficulties as $v_p(\dot{d})$ approaches zero. Interpreting the equation physically, it is plausible that $1 - \alpha \sim v_p$, so that Γ_2 vanishes at the limit. However, because α and \dot{d} (and, consequently, $v_p(\dot{d})$) are separate variables, their numerical values will, in general, approach the corresponding limits at different times and locations. In a computer program, the situation requires special safeguards to prevent overflow.

The special programming can be avoided if the number of particles is introduced as a dependent variable. This can be done by different approaches. In one approach, one assumes that the governing equations, Eqs. (4.1) for α and \dot{d} , and the source term correlation (4.3) hold exactly. Then the number of particles, $\dot{m}(t, x)$, can be introduced by a formal definition in terms of already defined functions. In a second approach, one avoids the use of the correlation (4.3) and defines $\dot{m}(t, x)$ concurrently with the particle volume function $v_p(\dot{d})$ such that the equation for α in the equation system (4.1) is satisfied approximately. Finally, one can define $\dot{m}(x, t)$ by a specific "reasonable" formula and then seek to determine a corresponding function $v_p(\dot{d})$ such that the equation for α is approximately satisfied. Each of the approaches requires some approximations. The last approach has the advantage that it provides guidelines how to choose the particle volume function $v_p(\dot{d})$.

We start with the first approach and define \dot{m} in terms of α and $v_p(\dot{d})$ as in Eq. (2.45) by

$$\dot{m}(t, x) = VG (1-\alpha) / v_p(\dot{d}) \quad . \quad (4.12)$$

The two governing equations for α and \dot{d} in Eqs. (4.1) are, if the definition of Γ_2 by Eq. (4.3) is used,

$$\frac{\partial(1-\alpha)}{\partial t} = - \nabla \cdot ((1-\alpha)\dot{u}) - (1-\alpha) \frac{s_p(\dot{d})}{v_p(\dot{d})} \langle \dot{d} \rangle$$

(4.13)

and

$$\frac{\partial \dot{d}}{\partial t} = - \dot{u} \cdot \nabla \dot{d} + \langle \dot{d} \rangle \quad .$$

Next, we express α in terms of \dot{m} and v_p using Eq. (4.12), and obtain

$$\alpha = 1 - \frac{\dot{m}}{VG} v_p(\dot{d}) \quad . \quad (4.14)$$

The expression (4.14) is substituted into the first Eq. (4.13). After simple manipulations, whereby the relation

$$\frac{dv_p(\dot{d})}{d\dot{d}} = - s_p(\dot{d}) \quad (4.15)$$

is assumed, one obtains from the system (4.13) the new system

$$\left. \begin{aligned} \frac{\partial \bar{m}}{\partial t} &= - \nabla \cdot (\bar{m} \bar{u}) , \\ \frac{\partial \bar{d}}{\partial t} &= - \bar{u} \cdot \nabla \bar{d} + \langle \dot{d} \rangle . \end{aligned} \right\} \quad (4.16)$$

Thus, one can replace the two governing equations (4.13) by the two equations (4.16) and the relation (4.14). If \bar{m} is used instead of α as dependent variable, then the source term Γ_2 in the equation system (4.1) is calculated by

$$\Gamma_2 = \frac{\bar{m}}{VG} s_p(\bar{d}) \langle \dot{d} \rangle , \quad (4.17)$$

instead of using Eq. (4.3). The expression (4.17) has no numerical singularities. In addition, the new Eqs. (4.16) are simpler than the previously used set (4.13). Physically interpreted, the first Eq. (4.16) means conservation of the number of particles, independently of their size, whereas the second equation governs the average size of the particles, independently of their number in the averaging volume.

The weak point of the described formal introduction of $\bar{m}(t, x)$ (the first approach) is that \bar{m} and the governing equation for \bar{m} contain inaccuracies that depend on the quality of the formula (4.3) for the source term Γ_2 . In order to make the definition of \bar{m} independent of these inaccuracies, one can define \bar{m} concurrently with $v_p(\bar{d})$ and $s_p(\bar{d})$ by the relation (4.12), which we rewrite in the form

$$\bar{m}(t, x) v_p(\bar{d}) = \int_V (1-\beta) g dV , \quad (4.18)$$

the Eq. (4.15), and

$$\bar{m}(t, x) s_p(\bar{d}) = \int_{S_p} g dS = SG . \quad (4.19)$$

The Eqs. (4.15), (4.18), and (4.19) are consistent in the sense that Eq. (4.19) is a consequence of Eqs. (4.15) and (4.18).

The exact expression for the source term Γ_2 is

$$\Gamma_2 = \frac{1}{VG} \int_{S_p} g \tilde{d} dS = \frac{SG}{VG} \tilde{d} . \quad (4.20)$$

Therefore, if Eq. (4.19) holds

$$\Gamma_2 = \frac{\bar{m}^*}{VG} s_p(\tilde{d}^*) \tilde{d} . \quad (4.21)$$

If we also use the exact average value \tilde{d} instead of the correlation $\langle \tilde{d} \rangle$ in the governing equation for \tilde{d} , then one obtains from these relations and from the last two Eqs. (4.1) by formal manipulation as above

$$\left. \begin{aligned} \frac{\partial \bar{m}^*}{\partial t} &= - \nabla \cdot (\bar{m}^* \mathbf{u}) , \\ \frac{\partial \tilde{d}^*}{\partial t} &= - \mathbf{u} \cdot \nabla \tilde{d}^* - \tilde{d} . \end{aligned} \right\} \quad (4.22)$$

Eqs. (4.21) and (4.22) are derived without any simplifying approximations for the source term. When the equations are incorporated into the equation system (4.1) for numerical solution, then the average \tilde{d} will, of course, be replaced by the corresponding correlation $\langle \tilde{d} \rangle$.

The weak point of the second approach is that the two functions \bar{m}^* and v_p with the desired properties do not exist in general, and, therefore, one has to use functions that satisfy the Eqs. (4.15), (4.18), and (4.19) only approximately. The non-existence can be seen, e.g., by considering the ratio s_p/v_p , which according to Eqs. (4.18) and (4.19) is equal to

$$s_p(\tilde{d}^*)/v_p(\tilde{d}^*) = \frac{\int_{S_p} g dS}{\int_V (1-\beta) g dV} . \quad (4.23)$$

The right-hand side of Eq. (4.23) obviously depends not only on the average \tilde{d} , but also explicitly on t and x . Even in the special case where all particles are equal, i.e., $\tilde{d} \equiv \tilde{d}^* = \text{constant}$, the ratio depends on the position of the grains, i.e., explicitly on t and x . On the other hand, if g is a constant, then Eq. (4.23) can be, indeed, a function of \tilde{d} only, and a proper function $v_p(\tilde{d})$ might be found. (Actually, g can be only approximately a constant, see Section 2.4.)

Because the Eqs. (4.15), (4.18), and (4.19) cannot be satisfied exactly, one might as well define, as a third approach, $\bar{m}(x, t)$ by a reasonable formula and then seek such a function $v_p(\bar{d})$ that satisfies the above mentioned equations approximately. (The other possibility, to choose $v_p(\bar{d})$ and then define \bar{m} by Eq. (4.18) amounts to the definition by Eq. (4.12). The corresponding \bar{m} has undesirable limit properties when some grains in the averaging volume are reduced by combustion to zero.)

Either of the following two formulas define functions $\bar{m}(t, x)$ with reasonable limit properties:

$$\bar{m}^* = \sum_{i=1}^n \left\{ \frac{1}{s_{pi}} \int_{S_i \cap V} g dS \right\} , \quad (4.24)$$

$$\bar{m}^* = \sum_{i=1}^n \left\{ \frac{1}{v_{pi}} \int_{V_i \cap V} g dV \right\} . \quad (4.25)$$

In these equations, n is the number of grains or grain parts in V , s_{pi} are the surface areas of the grains, S_i are their surfaces, v_{pi} are the magnitudes of their volumes, and V_i are their volumes. The contribution of a grain that is reduced to zero volume is $g(\xi_1(t) - x)$, where $\xi_1(t)$ is the location of the grain. When all grains are reduced to zero, then either of the formulas produces

$$\bar{m}(t, x) = \sum_{i=1}^n g(\xi_1(t) - x) . \quad (4.26)$$

If all grains have the same finite size, then the formulas reduce to Eqs. (4.18) and (4.19), respectively. Finally, if g is constant then the contribution to \bar{m} of each grain that is completely inside V is one, and the contribution of a grain partially in V is less than one, in accordance with its location. Only for constant g , and all grains located inside V , the function \bar{m} is independent of \bar{d} . Therefore, the factorization as postulated by Eqs. (4.18) and (4.19) can be best approximated if the weight function is constant over most of the averaging volume.

If \bar{m} is defined by either of the Eqs. (4.24) or (4.25), then one may select the volume correlation $v_p(\bar{d})$ to fit the choice of \bar{m} . The surface area correlation $s_p(\bar{d})$ is then obtained by the formula (4.15). The selection of $v_p(\bar{d})$ is discussed in Section 4.7.9.

4.2.2 Pressure Logarithm and Entropy. The equation system (4.1) contains two thermodynamic quantities as dependent variables, namely, the density ρ and the specific internal energy e . One can replace this pair of variables by a different pair of thermodynamic quantities and replace the first two equations in Eq. (4.1) by corresponding governing equations for the new pair. The variables can be chosen such that the new system of equations is better suited for numerical treatment.

First, we notice that up to six equations contain the gradient of the pressure. The handling of the gradient terms can be simplified considerably if the pressure p itself is chosen as a dependent variable instead of ρ . The replacement reduces the total number of terms in the equation system.

Second, one may replace e by another variable, e.g., by the specific entropy s , the specific enthalpy h , or the temperature T . These choices do not simplify the equations. The number of terms does not change if s is used instead of e , but it does increase if h is used instead of e . Choosing T as a dependent variable, one obtains the most complicated equations.

Based on these considerations, we have chosen s as a second thermodynamic variable. First, it does not complicate the equation system. Second, s is proportional to the logarithm of the temperature, whereas e is proportional to the temperature itself. Therefore, if the flow contains large temperature variations, its representation in terms of s is much smoother and more amenable to numerical differentiation. (One can expect large temperature variations in certain interior ballistics problems.)

The relation between s , p , and T is for Noble-Abel gases

$$s = A_1 \ln(T) - A_2 \ln(p) \quad (4.27)$$

with constant A_1 and A_2 . The Eq. (4.27) suggests that $q = \ln(p)$ would be an even better choice than p as the other thermodynamic variable. If q is a function of p only, then this replacement does not introduce any new complications in the governing equations. Our final choice of thermodynamic variables is, therefore, the specific entropy s [J/(kg·K)] and a pressure logarithm function q , which we define as

$$q(p) = q_1 [\ln(p/p_1) + 1] \quad , \quad (Pa) \quad (4.28)$$

with constant q_1 and p_1 .

The first two equations in the system of governing equations (4.1), if expressed in terms of s and q , are

$$\frac{\partial s}{\partial t} = -u \cdot \nabla s + \frac{p}{\rho T} B + H \Gamma + (\phi + \psi) \quad (4.29)$$

$$\frac{\partial q}{\partial t} = -u \cdot \nabla q - \frac{\rho}{\rho_q} \left[\nabla \cdot u + \frac{e_s}{T} B \right] + \frac{1}{e_q} [\hat{e} - e - e_s H] \Gamma - \frac{\rho_s}{\rho_q} (\phi + \psi) \quad ,$$

where

$$\begin{aligned}
 B &= \frac{1}{\alpha} [(1-\alpha) \nabla \cdot \mathbf{u}^* - (\mathbf{u} - \mathbf{u}^*) \cdot \nabla (1-\alpha)] , \\
 H &= \frac{1}{T} [(\hat{e} + p/\rho^*) - (e + p/\rho)] , \\
 \Gamma &= \frac{1}{\alpha} \frac{\rho^*}{\rho} \Gamma_2 = \frac{1}{\alpha} \frac{\rho^*}{\rho} \frac{m}{VG} s_p(\hat{d}) \langle \dot{d} \rangle , \\
 \Phi &= \frac{1}{T_0} \Phi_1 , \\
 \Psi &= \frac{1}{T_0} \Psi_1 ,
 \end{aligned} \tag{4.30}$$

and

$$\begin{aligned}
 \rho_q &= \frac{\partial \rho(p,s)}{\partial p} \frac{dp}{dq} , \\
 \rho_s &= \frac{\partial \rho(p,s)}{\partial s} , \\
 e_q &= \frac{\partial e(p,s)}{\partial p} \frac{dp}{dq} , \\
 e_s &= \frac{\partial e(p,s)}{\partial s} .
 \end{aligned} \tag{4.31}$$

In the derivation of the equation for q , we used the relationship $\rho^2 e_q = p \rho_q$ which can be obtained from the second law of thermodynamics (Hund, 1950). In Eq. (4.31), $dp/dq = p/q_1$ by Eq. (4.28), and the derivatives of the thermodynamic functions are modeled by the equation of state correlations, described in Section 4.7.1.

4.3 Final System of Governing Equations

The governing equations (4.1) can be expressed as follows in terms of the new set of variables that were introduced in Section 4.2.

$$\begin{aligned}
\frac{\partial s}{\partial t} &= -u \cdot \nabla s + \frac{p}{\rho T} B + H\Gamma + (\phi + \psi) , \\
\frac{\partial q}{\partial t} &= -u \cdot \nabla q - \frac{\rho}{\rho_q} (\nabla \cdot u + \frac{e_s}{T} B) + \frac{1}{e_q} (\hat{e} - e - e_s H) \Gamma - \frac{\rho_s}{\rho_q} (\phi + \psi) , \\
\frac{\partial u}{\partial t} &= - (u \cdot \nabla) u - \frac{p_q}{\rho} \nabla q - (u - \hat{u}) \Gamma - \frac{1-\alpha}{\alpha} A_{\text{drag}} + A_{\text{visc}} + A_{\text{turb}} , \\
\frac{\partial \hat{u}}{\partial t} &= - (\hat{u} \cdot \nabla) \hat{u} - \frac{p_q}{\rho} \nabla q + \frac{\rho}{\rho} A_{\text{drag}} + A_{\text{stress}} , \\
\frac{\partial \hat{m}}{\partial t} &= - \nabla \cdot (\hat{m} \hat{u}) , \\
\frac{\partial \hat{d}}{\partial t} &= - \hat{u} \cdot \nabla \hat{d} + \langle \dot{d} \rangle , \\
\frac{\partial \hat{T}}{\partial t} &= - \hat{u} \cdot \nabla \hat{T} + \langle \dot{T} \rangle
\end{aligned} \tag{4.32}$$

with

$$\begin{aligned}
B &= \frac{1}{\alpha} [(1-\alpha) \nabla \cdot \hat{u} - (u - \hat{u}) \cdot \nabla (1-\alpha)] , \\
\alpha &= 1 - \nabla_p (\hat{d}) \hat{m} / VG , \\
H &= \frac{1}{T} [(\hat{e} + p/\hat{\rho}) - (e + p/\rho)] , \\
\Gamma &= \frac{1}{\alpha} \frac{\rho}{\rho} \frac{\hat{m}}{VG} s_p (\hat{d}) \langle \dot{d} \rangle .
\end{aligned} \tag{4.33}$$

The partial derivatives ρ_s , ρ_q , e_s , and e_q are defined by Eq. (4.31). The derivative $p_q = dp/dq$ is equal to p/q_1 if q is the pressure logarithm defined by Eq. (4.28).

Models of the various correlation terms in Eq. (4.32) are discussed in Section 4.7. Their physical meaning is as follows: Γ represents the mass source due to combustion, ϕ represents the heat dissipation, ψ contains the heat conduction terms, $e(s,p)$, $T(s,p)$, and $\rho(s,p)$ are thermodynamic state functions, A_{drag} is the acceleration due to drag, A_{visc} is the acceleration due to viscosity, A_{turb} is the acceleration due to turbulence, A_{stress} is the

acceleration due to intergranular stress and solid phase turbulence, $\langle \dot{d} \rangle$ is the regression rate correlation, $\langle \dot{T} \rangle$ is a correlation for the heat conduction between gas and particles, \bar{e} is e at flame temperature, $s_p(\bar{d})$ is the average surface area of a single grain, and $v_p(\bar{d})$ is the average volume of a single grain. The variable $\langle \dot{T} \rangle$ enters also the first two equations (4.32) as an argument of the term γ .

The correlations are defined in terms of volume or surface averages. Therefore, the models of the correlations should be different for different averaging volumes and/or different weight functions. However, because experimentally determined correlation models are usually reported without reference to any averaging, their relation to specific averaging procedures are difficult to determine. Therefore, the influence of their relationship on the overall accuracy of the interior ballistics model has not been established.

4.4 Regions of Definition

According to Section 2.3, the average quantities describing gas properties are defined at all interior points of the gun tube, except for boundary regions the shape of which depends on the averaging volume. The average quantities are the density $\alpha\rho$, the energy density αe , and the momentum density vector αu . Consequently, all other quantities that are defined in terms of these quantities are defined in the same regions. Such quantities are, e.g., e , u , s , q , T , etc. The porosity α has the same region of definition. The grain number function \bar{m} also can be defined in the same region, if one uses the extension $\bar{m}^* = 0$ if the averaging volume contains no grains.

Average quantities describing grain properties are defined only at reference points for which the averaging volume contains grains. Therefore, the set of average conservation equations for $(1-\alpha)\rho u$, $(1-\alpha)\rho$, \bar{d} , and \bar{T} is not defined in regions without grains (see Section 3.3). By a reformulation of the conservation equations, we obtained in Section 4.3 an equivalent set of governing equations (4.32). This set has no singularities at $\alpha = 1$ and it enables one to calculate nominal grain properties at all interior points where the gas properties are defined. Therefore, one can extend the definition of average grain properties as follows. The grain properties are defined by the averaging integrals (see Section 2.2), if the averaging volume contains grains. In other regions, the grain properties are defined as the solution of Eqs. (4.32). In interior ballistics problems this definition amounts to an interpolation of \bar{u} , \bar{d} , and \bar{T} across regions without grains. When the grains have been reduced to zero volume, one can still calculate their motion, which now corresponds to a so-called "dusty gas" model. In such a gas, the dust follows the gas flow according to a drag law, but it does not influence the gas flow itself. Using the set (4.32) as governing equations one obtains

regions of "dusty gas" where $\bar{m}^* > 0$ and $v_p(\bar{d}) = 0$. In regions with $\bar{m}^* = 0$ and $v_p(\bar{d}) > 0$ the equations provide an interpolation of \bar{u} , \bar{t} , and \bar{d} in space and time between regions with grains.

In the boundary regions discussed in Section 2.3, none of the average quantities are defined and, consequently, the differential eqs. (4.32) have no meaning in these regions. Strictly speaking, one should provide boundary conditions for Eqs. (4.32) at the boundaries $l/2$ away from the tube walls and $l/2$ or $l/3$ away from the breech and projectile, if the average volume is defined as a sphere (2.47) or cylinder (2.49). The meaning of the solution of the equations in the boundary regions is not obvious if one prescribes boundary conditions on the solid boundaries instead. Section 4.6 contains a discussion of the boundary condition problems.

4.5 Initial Conditions

Typical local initial conditions for interior ballistics problems are constant state conditions over the entire region. Because averaging of a constant produces the same constant, the initial averages in most cases are simply equal to the local values.

Deviation from a constant initial state typically involves either a porosity α that is not uniform, or a non-uniform grain size, i.e., a non-uniform \bar{d} . In these cases, one cannot use the local values of \bar{m} and \bar{d} as initial values. Instead, the initial profiles must be computed by averaging the local values, whereby the same averaging volume V and weight function g are used as for the correlation models and boundary conditions.

In regions where initially the grain number \bar{m} is zero one has to extrapolate or interpolate the values of \bar{u} , \bar{d} , and \bar{t} . The initial grain velocity is normally identically zero and one can use $\bar{u} = 0$ for the extrapolation. Likewise, the initial grain surface temperature is usually constant, and the same constant can be used for extrapolation. The regression distance may not be constant if different sizes of grains are loaded in different regions. In such cases, one has to use a common sense extrapolation that produces a smooth initial surface $\bar{d}(0,x)$.

In the boundary regions, "correct" initial values cannot be specified for reasons explained in Section 4.4. The proper choice of these initial values depends on the method of treatment of the boundary regions. However, one can assume that any reasonable treatment will produce uniform values, if the local function values are uniform. Therefore, one may specify in the boundary regions the same uniform initial values as in the interior region. If the initial conditions are not uniform, then one has to design such an extrapolation of the averages to the boundary that is consistent with the treatment of boundary conditions.

4.6 Boundary Conditions

A theory that could provide guidelines for the formulation of boundary conditions for averaged equations has not been developed. Therefore, interior ballistics calculations usually are done with plausible ad hoc assumptions about boundary values. In this section we shall outline the requirements for a boundary condition theory and suggest a possible approach to the formulation of such a theory. Because the theory has not been developed, we shall also discuss ad hoc boundary conditions.

Discussing boundary conditions for averaged differential equations in confined volumes, we have to distinguish between two boundaries. For the purpose of the present discussions, we call them the outer boundary and the inner boundary, respectively. The outer boundary consists of the solid walls of the volume. In interior ballistics the solid walls are the tube walls, the breech, and the base of the projectile. The inner boundary is the limit of validity of the averaged differential equations. As discussed in Sections 2.2 and 4.4, the inner boundary is located a finite distance inward from the outer boundary. The magnitude of the distance depends on the size of the averaging volume. If the averaging volume is a sphere with the diameter l , then the inner boundary is located $l/2$ away from the tube walls, breech, and projectile. If the averaging volume is the cylinder described in Section 2.2, then the inner boundary is $l/2$ away from the tube walls and $l/3$ away from the breech and projectile base. Let the region between the outer and inner boundaries be called the boundary region, and the region inside the inner boundary be called the interior region.

Classical theory for the discussion of necessary boundary conditions, well-posedness, and existence can be only applied to the inner boundary. Gough (1974) presents some of the discussion, implicitly assuming that the conditions on both boundaries are identical. The assumption is permissible if the size of the boundary region is small compared to the size of salient structures of the flow field. Because the size of the boundary region must be large compared to the size of propellant grains (see Section 2.2), it is generally not small compared to, e.g., the gas boundary layer. For interior ballistics flows, therefore, one cannot assume that boundary conditions on the inner and outer boundaries are identical.

Physical boundary conditions, such as $u = u_{\text{wall}}$, are only given for the local gas phase functions on the outer boundary. The only physical boundary condition for the particles is that no single particle can penetrate the wall. In addition, one may also formulate collision conditions for single particles impacting on the wall, i.e., on the outer boundary.

A boundary condition theory for averaged equations has to bridge the gap between the outer and inner boundaries. It should provide a complete set of boundary conditions for the average quantities on the inner boundary in terms of the local physical boundary conditions on the outer boundary.

One possible approach to the problem is by construction of a continuation of the solution into the boundary region. If such a continuation is established, then one has reduced the problem to the formulation of boundary conditions on the outer boundary only. The simplest method to obtain a continuation is to define it as the solution of the same differential equations that are valid in the interior region. Then one needs only conditions on the outer boundary and disregards the existence of the inner boundary. This is the usual approach in two-phase flow calculations. It has the deficiency that one has no guidelines how to formulate the boundary conditions for the continued functions, because they are neither the local functions nor the average functions.

A more promising continuation may be obtained by changing the definition of the averages such that it includes the boundary region. This requires that the averaging volume V has a shape that depends on the position x of the reference point. The conservation equations of Section 3 are derived under the assumption of a fixed size and shape of V . The averages defined for a variable V satisfy a different set of differential equations. The continuation into the boundary region could be computed by solving Eqs. (4.32) in the interior region and the new set in the boundary region, and by matching both solutions at the inner boundary. The boundary conditions on the outer boundary then represent conditions for averaged functions and can be modeled accordingly.

Because a theory of the described type is not available, we now formulate ad hoc conditions that may be used for the differential equation system (4.32).

The local boundary conditions for the gas are: $u = u_{\text{wall}}$, a condition for the temperature prescribing either $T = T_{\text{wall}}$ or $\partial T / \partial n = (\partial T / \partial n)_{\text{wall}}$, where n is the normal to the wall, and the mass conservation equation. In the spirit of interpreting the solutions of the differential equations as averages, one would not directly use these conditions as boundary conditions. Instead, some interpolation is needed that reflects the averaging. We propose the following approach.

Let $\lambda/2$ be the distance between the inner and outer boundary and let ϵ be the thickness of the gas boundary layer. Let ϕ be a function with prescribed local boundary value ϕ_{wall} and n_1 be the unit normal to the inner boundary, pointing outward with respect to the interior. We then use the following boundary condition on the outer boundary for gas properties

$$\phi_{\text{outerb}} = \left[\frac{\lambda}{2} (\phi_{\text{innerb}} + (\nabla \phi_{\text{innerb}} \cdot n_1) \frac{\lambda}{2}) + \epsilon \phi_{\text{wall}} \right] / \left(\frac{\lambda}{2} + \epsilon \right) \quad (4.34)$$

Because λ is larger than a particle diameter (see Section 2.3), the boundary value on the outer boundary, when computed by Eq. (4.34), will approach the local boundary value only if the particles are small compared to the thickness

of the boundary layer ($\epsilon \gg l/2$). This may be the case, e.g., when the flow of wear reducing additives is investigated. If the particles are large compared to the thickness of the boundary layer ($l/2 \gg \epsilon$), then the outer boundary value given by Eq. (4.34) approaches an extrapolated value from the inner boundary.

Eq. (4.34) may be used to determine the boundary values of u , and T or $\partial T/\partial n$. The average gas continuity equation may be used to close the set of boundary conditions for gas properties.

The formulation of a boundary condition for the average particle velocity presents a dilemma. On one hand, the condition should prevent the particles from penetrating the wall. On the other hand, the average particle velocity at the outer boundary may very well point into the wall, merely indicating an accumulation of particles within the averaging volume. As an ad hoc measure, we disregard the second possibility and suggest for the average particle velocity at the outer boundary the following formula. Let \tilde{u}_{DE}^* be the solution obtained from the differential equation system (4.32) at the outer boundary, \tilde{u}_{wall} be the velocity of the wall, and n_{wall} be the unit normal to the wall pointing outward. Then the outer boundary value of \tilde{u} is

$$\tilde{u}_{outerb}^* = \tilde{u}_{DE}^* - n_{wall} \max(0, (\tilde{u}_{DE}^* - \tilde{u}_{wall}) \cdot n_{wall}) \quad (4.35)$$

The resulting \tilde{u}_{outerb}^* satisfies the condition

$$(\tilde{u}_{outerb}^* - \tilde{u}_{wall}) \cdot n_{wall} \leq 0 \quad (4.36)$$

which prevents the particles from flowing through the wall.

The quantities \tilde{m} , \tilde{d} , and \tilde{T} are computed by solving the corresponding governing equations at the outer boundary.

4.7 Models of Correlations

4.7.1 Equations of State. For the derivation of the average equations in Section 3, we used the averages of two thermodynamic quantities, namely, the density ρ and the specific internal energy e . The conservation equations contain two other thermodynamic quantities, the pressure p , and the temperature T . (The latter enters the heat conduction term and may be also used in other correlations.) They were assumed to be related to e and ρ by equations of state, i.e., by

$$p = p(\rho, e)$$

and

$$T = T(\rho, e) .$$

(4.37)

Generally, one uses, in Eq. (4.37), the same functions that hold locally. This introduces errors in several terms of the average conservation equations.

As an example, let us consider the error term in the average momentum equation. The error made by approximating the volume average of the local pressure by the first equation in (4.37) is from Eq. (3.31)

$$C_m = \frac{1}{\alpha \rho} \nabla [\alpha(t, x) (\tilde{p}(t, \hat{x}) - \tilde{p}(t, x))] . \quad (4.38)$$

As discussed in Section 3.2.2, to minimize the error by a proper choice of the function p , we need to minimize the errors in the functional values as well as in the gradient values. However, the pressure function enters the equation system in various places and different combinations. Therefore, the use of the local equations of state is probably as good as approximation as any. Correspondingly, one also uses the local equations of state when the entropy s is introduced as a dependent variable.

All thermodynamic variables (temperature, pressure, density, energy, entropy, and enthalpy) are completely determined in terms of two variables if two "equations of state" are provided by postulate or measurement. Using the two given equations, all other relations can be derived from the laws of thermodynamics, which provide the following three systems of differential equations (Bund, 1950):

$$\begin{aligned} \frac{\partial c_v(\rho, T)}{\partial \rho} &= - \frac{1}{\rho^2} T \frac{\partial^2 p(\rho, T)}{\partial T^2} , \\ c_p - c_v &= \frac{T}{\rho^2} \left(\frac{\partial p(\rho, T)}{\partial T} \right)^2 / \left(\frac{\partial p(\rho, T)}{\partial \rho} \right) . \end{aligned} \quad (4.39)$$

(c_p and c_v are the specific heats (J/(kg·K)) for constant pressure and volume, respectively),

$$\left. \begin{aligned} \frac{\partial e(\rho, T)}{\partial \rho} &= -\frac{1}{\rho^2} \left(T \frac{\partial p(\rho, T)}{\partial T} - p \right) , \\ \frac{\partial e(\rho, T)}{\partial T} &= c_v , \end{aligned} \right\} (4.40)$$

and

$$\left. \begin{aligned} \frac{\partial s(\rho, T)}{\partial \rho} &= -\frac{1}{\rho^2} \frac{\partial p(\rho, T)}{\partial T} , \\ \frac{\partial s(\rho, T)}{\partial T} &= \frac{1}{T} c_v . \end{aligned} \right\} (4.41)$$

An equation of state that is often used in interior ballistics is the Noble-Abel equation

$$p(\rho, T) = \frac{R}{M} T \frac{\rho}{1 - \eta \rho} , \quad (4.42)$$

where $R = 8.3143 \text{ J/(mol}\cdot\text{K)}$ is the universal gas constant, $M \text{ (kg/mol)}$ is the molar mass, and $\eta \text{ (m}^3\text{/kg)}$ is the covolume. From Eqs. (4.39) and (4.42) one finds that for a Noble-Abel gas

$$\left. \begin{aligned} c_v &= c_v(T) \\ \text{and} \\ c_p &= c_v(T) + \frac{R}{M} . \end{aligned} \right\} (4.43)$$

Therefore, in order to completely specify the gas, one has to provide, in addition to Eq. (4.42), a temperature function $c_v(T)$. Alternatively one can specify instead of $c_v(T)$ a function $c_p(T)$, or a function $\gamma(T)$ that gives the ratio $c_p/c_v = \gamma(T)$. In the latter case, the specific heat functions are

$$\left. \begin{aligned} c_v(T) &= \frac{1}{\gamma(T) - 1} \frac{R}{M} \\ \text{and} \\ c_p(T) &= \frac{\gamma(T)}{\gamma(T) - 1} \frac{R}{M} . \end{aligned} \right\} (4.44)$$

We assume that $\gamma(T)$ is constant, and obtain with this assumption the functions $e(\rho, T)$ and $s(\rho, T)$ by integration of Eqs. (4.40) and (4.41). After some manipulations, one can express the quantities of interest in terms of p and s , as required by the system of governing equations (4.32). The results are listed below. T_R and p_R are reference values which determine the integration constant for the entropy.

$$\begin{aligned} T(p, s) &= T_R \left(\frac{p}{p_R} \right)^{\frac{\gamma-1}{\gamma}} \exp \left(\frac{M}{R} \frac{\gamma-1}{\gamma} s \right), \quad K, \\ e &= \frac{1}{\gamma-1} \frac{R}{M} T, \quad J/kg, \\ \rho &= \left(\frac{R}{M} \frac{T}{p} + \eta \right)^{-1}, \quad kg/m^3, \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} (4.45)$$

$$\begin{aligned} \frac{\partial T(p, s)}{\partial p} &= \frac{\gamma-1}{\gamma} \frac{T}{p}, \\ \frac{\partial T(p, s)}{\partial s} &= \frac{\gamma-1}{\gamma} \frac{M}{R} T, \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} (4.46)$$

$$\begin{aligned} \frac{\partial e(p, s)}{\partial p} &= \frac{\gamma-1}{\gamma} \frac{e}{p}, \\ \frac{\partial e(p, s)}{\partial s} &= \frac{1}{\gamma} T, \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} (4.47)$$

and

$$\begin{aligned} \frac{\partial \rho(p, s)}{\partial p} &= \frac{1}{\gamma} \frac{\rho}{p} (1 - \eta \rho), \\ \frac{\partial \rho(p, s)}{\partial s} &= - \frac{M}{R} \frac{\gamma-1}{\gamma} \rho (1 - \eta \rho). \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} (4.48)$$

The square of the sound speed is

$$a^2 = \gamma \frac{p}{\rho} \frac{1}{1 - \eta \rho}, \quad m^2/s^2. \quad (4.49)$$

The specific entropy, expressed in terms of pressure and temperature, is

$$s = \frac{R}{M} \frac{1}{\gamma-1} \ln \left[\frac{T}{T_R} \left(\frac{P}{P_R} \right)^{\frac{1-\gamma}{\gamma}} \right] , \quad J/(kg \cdot K) . \quad (4.50)$$

4.7.2 Acceleration by Gaseous Stresses. The governing equation for the average gas velocity in the equation system (4.32) contains the terms A_{visc} and A_{turb} . The former term represents the acceleration due to laminar viscosity. The latter term represents the acceleration due to turbulence. A simple turbulence model is a Reynolds stress model with viscosity coefficients depending, e.g., on temperature. Then the forms of A_{visc} and A_{turb} are identical. We restrict our discussion to the term A_{visc} . More complicated turbulence models are possible (see Gibeling et al., 1980), but will not be discussed in this report.

According to Section 3.3, the viscous acceleration term is

$$A_{visc} = \frac{1}{\alpha \rho} \nabla \cdot (\alpha \tilde{\Pi}) , \quad (4.51)$$

where $\tilde{\Pi}$ models the gas volume average of the local viscous stress tensor $\tilde{\Pi}$. The local tensor is given in terms of the strain rate tensor \tilde{E} by (Tsien, 1958, p. 13)

$$\tilde{\Pi} = 2\tilde{\mu} \tilde{E} + \left(\tilde{\lambda} - \frac{2}{3} \tilde{\mu} \right) \text{trace}(\tilde{E}) \mathbf{I} , \quad (4.52)$$

where $\tilde{\mu}$ and $\tilde{\lambda}$ are the shear viscosity coefficient and the bulk viscosity coefficient, respectively. Both are assumed to be functions of temperature. The strain rate tensor is defined by

$$\tilde{E} = \frac{1}{2} (\nabla \tilde{u} + (\nabla \tilde{u})^T) . \quad (4.53)$$

The modeling of the average viscous acceleration term involves models of the average viscosity coefficients and a model of the average strain rate tensor \tilde{E} .

The models of the average viscosity coefficients are purely empirical. A convenient set of formulas is the following generalization of the so-called Sutherland formula:

$$\begin{aligned}
 \mu(T) &= \mu_0 + \mu_1 \frac{T^{1.5}}{\mu_2 + T} , & \text{Pa}\cdot\text{s} , \\
 \text{and} \\
 \lambda(T) &= \lambda_0 + \lambda_1 \frac{T^{1.5}}{\lambda_2 + T} , & \text{Pa}\cdot\text{s} .
 \end{aligned}
 \tag{4.54}$$

The generalization consists of the addition of the parameters μ_0 and λ_0 , thereby including in the model the constant functions.

The average strain rate tensor \bar{E} is usually modeled by applying the local formula (4.53) to the average velocities. Then Π is obtained by using Eqs. (4.52) without the tildes and (4.54) with temperature $T(\rho, e)$ calculated from the average values of ρ and e . The approximation error is Eq. (3.32) divided by $\alpha\rho$, i.e.,

$$C_m = \frac{1}{\alpha\rho} \nabla \cdot \int_V \beta g [\tilde{\Pi}(\tilde{u}, \tilde{\rho}, \tilde{e}) - \Pi(u, \rho, e)] dV . \tag{4.55}$$

The error part that comes from the replacement of $\tilde{\rho}$ and \tilde{e} by ρ and e is probably smaller than the uncertainties of the empirical formula (4.54). However, the error part that comes from the use of the average velocity in Eq. (4.53), can be large because the formula involves derivatives of the velocity and in a viscous two-phase flow the local derivatives can be quite large. It is not necessary that the integration (4.55) cancels out locally large undulations of the integrand. An empirical correction based on careful experiments certainly could enhance the usefulness of the described model of the viscous acceleration term.

4.7.3 Heat Dissipation. All the heat dissipation terms are denoted by ϕ and they enter the governing Eqs. (4.32) for the specific entropy s and for the pressure logarithm function q . According to Sections 3.2.3, 3.3, 4.1, 4.2, and 4.3 the term ϕ models

$$\frac{1}{\alpha\rho T} \frac{1}{VG} \int_V \beta g \tilde{\phi} dV , \tag{4.56}$$

where the local heat dissipation function $\tilde{\phi}$ is given by (Tsien, 1958, p. 15)

$$\tilde{\phi} = 2 \tilde{\mu} \text{trace} (\tilde{E}^2) + (\tilde{\lambda} - \frac{2}{3} \tilde{\mu}) (\text{trace} \tilde{E})^2 , \quad W/m^3 , \tag{4.57}$$

\tilde{E} is the local strain rate tensor, and $\tilde{\mu}$ and $\tilde{\lambda}$ are the shear and bulk viscosity coefficients, respectively.

Usually $\bar{\phi}$ is defined in the same fashion as the equations of state (Section 4.7.1), i.e., by calculating a $\bar{\phi}$ with the same formula as $\tilde{\phi}$, but using the average velocities instead of the local velocities. The modeling of the viscosity coefficients is discussed in Section 4.7.2. In Cartesian coordinates, the formula is (Tsien, 1958, p. 15)

$$\frac{1}{\rho T} \bar{\phi}(E) = \frac{1}{\rho T} \left[\frac{1}{2} \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)^2 + \left(\lambda - \frac{2}{3} \mu \right) \left(\frac{\partial u_i}{\partial x_i} \right)^2 \right] , \quad (4.58)$$

whereby summation over i and j is assumed.

Even without considering turbulence, Eq. (4.58) likely underestimates the value of the expression (4.56) because local undulations will generally increase the value of the integrand. If a difference exists between the average velocities of the phases, then the local velocity gradients are particularly large.

In order to estimate their effect, we compute the heat dissipation term in a linear flow field superposed by an undulation. Particularly, we assume the following velocity components in Cartesian coordinates:

$$\begin{aligned} u_1 &= U + \frac{\Delta u}{L} x + \hat{u}(x, y, z) , \\ u_2 &= \hat{u}(x, y, z) , \\ u_3 &= \hat{u}(x, y, z) , \end{aligned} \quad \left. \vphantom{\begin{aligned} u_1 &= U + \frac{\Delta u}{L} x + \hat{u}(x, y, z) , \\ u_2 &= \hat{u}(x, y, z) , \\ u_3 &= \hat{u}(x, y, z) , \end{aligned}} \right\} (4.59)$$

where

$$\hat{u}(x, y, z) = \hat{U} \sin \left(\frac{2\pi}{L} x \right) \sin \left(\frac{2\pi}{L} y \right) \sin \left(\frac{2\pi}{L} z \right) . \quad (4.60)$$

The local heat dissipation for this flow field is

$$\begin{aligned} \tilde{\phi} &= \frac{1}{\rho T} \left[\frac{1}{2} \mu (\phi_{xx}^2 + \phi_{yy}^2 + \phi_{zz}^2 + 2\phi_{xy}^2 + 2\phi_{xz}^2 + 2\phi_{yz}^2) \right. \\ &\quad \left. + \left(\lambda - \frac{2}{3} \mu \right) \frac{1}{4} (\phi_{xx} + \phi_{yy} + \phi_{zz})^2 \right] , \end{aligned} \quad (4.61)$$

where

$$\begin{aligned}
 \phi_{xx} &= \hat{U} \frac{2\pi}{L} 2 \cos\left(\frac{2\pi}{L} x\right) \sin\left(\frac{2\pi}{L} y\right) \sin\left(\frac{2\pi}{L} z\right) + 2 \frac{\Delta u}{L}, \\
 \phi_{yy} &= \hat{U} \frac{2\pi}{L} 2 \sin\left(\frac{2\pi}{L} x\right) \cos\left(\frac{2\pi}{L} y\right) \sin\left(\frac{2\pi}{L} z\right), \\
 \phi_{zz} &= \hat{U} \frac{2\pi}{L} 2 \sin\left(\frac{2\pi}{L} x\right) \sin\left(\frac{2\pi}{L} y\right) \cos\left(\frac{2\pi}{L} z\right), \\
 \phi_{xy} &= \hat{U} \frac{2\pi}{L} \sin\left(\frac{2\pi}{L} (x+y)\right) \sin\left(\frac{2\pi}{L} z\right), \\
 \phi_{xz} &= \hat{U} \frac{2\pi}{L} \sin\left(\frac{2\pi}{L} (x+z)\right) \sin\left(\frac{2\pi}{L} y\right), \\
 \phi_{yz} &= \hat{U} \frac{2\pi}{L} \sin\left(\frac{2\pi}{L} x\right) \sin\left(\frac{2\pi}{L} (y+z)\right).
 \end{aligned}
 \tag{4.62}$$

Next we assume that the averaging volume is a cube with side lengths nL and that the weight function g is constant. For that case, the integral (4.56) yields

$$\phi = \frac{1}{\rho T} \left[\left(\frac{\Delta u}{nL} \right)^2 \left(\frac{4}{3} \mu + \lambda \right) + \left(\frac{\hat{U}}{L} \right)^2 \pi^2 \left(5\mu + \frac{3}{2}\lambda \right) \right]. \tag{4.63}$$

The first term in the brackets in Eq. (4.63) is the contribution of the linear field to ϕ . The second term is the contribution of the superposed undulations. One sees that for $\Delta u/(nL) = U/L$ the contribution of the undulations is about 40 times larger than that of the linear flow field. Interestingly, the contribution of the undulations does not depend on the number of periods in the averaging volume, but only on the amplitude and wave length. The example shows that the usual approximation of ϕ by the formula (4.58) can be grossly in error.

A model of the contributions of undulations in two-phase flow due to the difference between u and \bar{u} can be derived in the same manner as Eq. (4.63). To simplify the formulas let us choose the coordinate system such that the x -direction coincides with the direction of $u - \bar{u}$. Then the velocity undulations may be approximated by

$$\left. \begin{aligned} \hat{u}_1 &= (u-u^*) \sin\left(\frac{2\pi}{D}x\right) \sin\left(\frac{2\pi}{D}y\right) \sin\left(\frac{2\pi}{D}z\right) , \\ \hat{u}_2 &= 0 , \\ \hat{u}_3 &= 0 , \end{aligned} \right\} (4.64)$$

where D is the distance between the centers of the particles.

Let n be the number of particles in the averaging volume. We associate each maximum of the function u_1 with a particle. Then there are four particles in an elemental volume D^3 and $n = 4V/D^3$. Therefore,

$$D = (4V/n)^{1/3} . \quad (4.65)$$

The contribution of the undulations (4.64) to the dissipation function is one third of the contribution of the undulations (4.59) in all velocity coordinates, as can be verified. Therefore, a reasonable model for the contribution due to velocity differences is

$$\langle \phi \rangle = \frac{1}{\rho T} (u-u^*)^2 \left(\frac{n}{4V}\right)^{2/3} \pi^2 \left(\frac{5}{3} \mu + \frac{1}{2} \lambda\right) , \quad W/(kg \cdot K) . \quad (4.66)$$

In a computer program, where n and V are not available, one can use in Eq. (4.66) the quotient $\bar{m}/(VG)$ instead of n/V without changing the magnitude of $\langle \phi \rangle$. The correlation (4.66) probably gives only the order of magnitude of the contribution due to velocity differences in the flow. However, it certainly is better than the usual assumption $\langle \phi \rangle = 0$. In relation to the error term involving the dissipation function in Eq. (3.51), the function $\langle \phi \rangle$ approximates the error between the volume average of the local dissipation function and the average dissipation function $\bar{\phi}(E)$.

The models for the turbulent dissipation function varies widely. A simple model for ϕ_T is one which has an identical form to $\bar{\phi}$ (Eq. (4.58)) but with different viscosity coefficients. Gibeling et al. (1980), suggest a model based on an algebraic relationship among a turbulent length scale, turbulent viscosity, and turbulent kinetic energy.

The complete dissipation term that enters the governing equations is the sum of Eqs. (4.58), (4.66), and the model for ϕ_T :

$$\phi = \frac{1}{\rho T} \bar{\phi}(E) + \langle \phi \rangle + \frac{1}{\rho T} \phi_T , \quad W/(kg \cdot K) . \quad (4.67)$$

The approximation error is the difference between the expressions (4.56) and (4.67).

4.7.4 Heat Conduction. The heat conduction term ∇ enters the governing equation, Eq. (4.32), in two places. The term itself models at least two phenomena: the heat conduction within the gas defined in terms of the average quantities, and the heat conduction from the gas to the solid. Depending on the model for the fluctuations of ρ_{eu} from ρ_{eu} , we also can have a turbulent heat flux vector defined in a similar manner as the average heat conduction. We shall discuss each of these models in turn.

Locally, the heat conduction within the gas is assumed to be governed by Fourier's law

$$\tilde{Q} = -\tilde{\kappa}(\tilde{T})\nabla\tilde{T} \quad , \quad W/m^2 \quad , \quad (4.68)$$

where $\tilde{\kappa}(\tilde{T})$ is the thermal conductivity coefficient which depends on the local temperature. The corresponding average heat conduction term in Eq. (4.32) is a model of

$$-\frac{1}{\alpha\rho T} \nabla \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{Q} dV \right] = \frac{1}{\alpha\rho T} \nabla \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{\kappa}(\tilde{T}) \nabla \tilde{T} dV \right] \quad . \quad (4.69)$$

The volume average in expression (4.69) is usually modeled as Eq. (4.68) without the tildes, that is, the average value of temperature T (obtained from the average values of s and q by the equation of state correlations, Section 4.7.1), replacing the local temperature \tilde{T} and an average thermal conductivity coefficient κ replacing the local coefficient $\tilde{\kappa}$. The average thermal conductivity coefficient can be modeled by a generalized Sutherland-type correlation,

$$\kappa(T) = \kappa_0 + \kappa_1 \frac{T^{1.5}}{\kappa_2 + T} \quad , \quad W/(m \cdot K) \quad . \quad (4.70)$$

An estimate of the error incurred by using the model instead of expression (4.69) can be obtained as follows when V_{gas} is connected:

$$\begin{aligned} C_Q &= \frac{-1}{\alpha\rho T} \nabla \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{Q} dV - \alpha Q \right] = \frac{1}{\alpha\rho T} \nabla \cdot \left[\frac{1}{VG} \int_V \beta g \tilde{\kappa} \nabla \tilde{T} dV - \alpha \kappa \nabla T \right] = \\ &= \frac{1}{\alpha\rho T} \nabla \cdot \left[\hat{\alpha} \hat{\kappa} \nabla \hat{T} - \alpha \kappa \nabla T \right] \quad , \end{aligned} \quad (4.71)$$

where $\hat{T} = \tilde{T}(\hat{s}, \hat{q})$ and $\hat{k} = \tilde{k}(\hat{T})$ are mean value points of the integrand. Expanding Eq. (4.71) further one obtains

$$C_Q = \frac{1}{\alpha \rho T} \nabla \cdot [\alpha(\tilde{\kappa} - \kappa) \nabla \hat{T} + \alpha \kappa \nabla (\hat{T} - T)] \quad (4.72)$$

and

$$|C_Q| < \max_V \left| \frac{\beta}{\alpha \rho T} \nabla \cdot [\alpha(\tilde{\kappa} - \kappa) \nabla \hat{T} + \alpha \kappa \nabla (\hat{T} - T)] \right| \quad (4.73)$$

The term involving the difference $\tilde{\kappa} - \kappa$ can be reduced if the coefficients κ_0 , κ_1 , and κ_2 in the correlation (4.70) are chosen such that

$$\kappa(T) = \frac{1}{\alpha V G} \int_V \beta g \tilde{\kappa} dV \quad (4.74)$$

The term involving $\nabla(\hat{T} - T)$ reflects the modeling error due to local undulations of the gas temperature.

The heat conduction between the gas and the particles is represented in Eq. (4.32) by a model of

$$-\frac{1}{\alpha \rho T} \frac{1}{V G} \int_{S_p} g \tilde{Q} \cdot n_{sp} dS = \frac{1}{\alpha \rho T} \frac{1}{V G} \int_{S_p} g \tilde{\kappa} \nabla \hat{T} \cdot n_{sp} dS \quad (4.75)$$

The integrand in Eq. (4.75) is the heat flux into the particles. We define the surface averaged heat flux by

$$\dot{e} = -\frac{1}{S G} \int_{S_p} g \tilde{\kappa} \nabla \hat{T} \cdot n_{sp} dS, \quad W/m^2, \quad (4.76)$$

and rewrite expression (4.75) as

$$-\frac{1}{\alpha \rho T} \frac{S G}{V G} \dot{e} \quad (4.77)$$

The quantity \dot{e} is modeled by experimental correlations which can have various different forms. A relatively simple formula is (Gibeling et al., 1980)

$$\langle \dot{e} \rangle = \frac{\dot{m}}{S G} s_p [h_c(T - \hat{T}) + h_r(T - \hat{T})], \quad W/m^2, \quad (4.78a)$$

where \bar{T} is the average grain surface temperature. The coefficients h_c and h_r in Eq. (4.78) model the heat transfer by conduction and radiation, respectively. Gibeling et al. (1980), suggest the following expression for the coefficients in case of spherical particles and Noble-Abel gas:

$$h_c = \frac{\kappa}{\bar{D}_p/2} + 0.2 \left(\frac{\gamma}{\gamma-1} \frac{R}{M} \frac{(\kappa_2 p)^2 |u-\bar{u}|^2}{\mu \bar{D}_p/2} \right)^{1/3}, \quad W/(m^2 K), \quad (4.78b)$$

where \bar{D}_p is the diameter of the particles, and μ is the shear viscosity coefficient (Section 4.7.2), and

$$h_r = \epsilon \sigma_{SB} (T+\bar{T}) (T^2+\bar{T}^2), \quad W/(m^2 K), \quad (4.78c)$$

where ϵ is the particle emissivity and $\sigma_{SB} = 5.67032 \cdot 10^{-8} W m^{-2} K^{-4}$ is the Stephan-Boltzmann constant.

The model $\langle \dot{e} \rangle$ should be consistent with the model $\langle \dot{T} \rangle$ of the grain surface temperature rate of change. The relation between both models is discussed in Section 4.7.10.

The model of the significant fluctuations of $\bar{p}eu$ from p_{eu} (denoted by Q_T , see Section 3.3) can have different forms. One model of the turbulent heat flux vector, given by Ishii (1975) and Gibeling et al. (1980), is

$$Q_T = -\kappa_T \left[\nabla T - \frac{\nabla \alpha}{\alpha} (T_1 - T) \right], \quad W/m^2, \quad (4.79)$$

where T_1 is an average temperature on the interface (a function of T and \bar{T}) and κ_T is given by an algebraic formula involving an effective viscosity and Prandtl number.

The heat conduction term Ψ is the sum of the three described models, i.e.,

$$\Psi = \Psi_{gas} + \Psi_{particle} + \Psi_{turb} = \quad (4.80)$$

$$= \frac{1}{\alpha \rho T} \nabla \cdot (\alpha \kappa \nabla T) - \frac{1}{\alpha \rho T} \frac{SG}{VG} \langle \dot{e} \rangle - \frac{1}{\alpha \rho T} \nabla \cdot Q_T, \quad W/(kg \cdot K).$$

4.7.5 Acceleration by Drag. The acceleration by drag between gas and particles enters the governing equations (4.32) for the velocities u and u^* . The term is defined by (Section 3.3.)

$$A_{\text{drag}} = \frac{1}{(1-\alpha)\rho} \frac{1}{VG} D \quad , \quad (4.81)$$

where D models

$$\frac{1}{VG} \int_{S_p} g[n_{sp}(\tilde{p} - p) - n_{sp} \cdot \tilde{\Pi}] dS \quad , \quad (4.82)$$

\tilde{p} and $\tilde{\Pi}$ are the local pressure and viscous stress tensor, and p is the average pressure. In interior ballistics applications, the term is modeled by experimental correlations that are available for single particles (e.g., spheres) and for packed beds of particles. For situations between these extremes one has to interpolate.

In order to see how the drag coefficient c_D for a single sphere relates to A_{drag} , we consider a situation where the $\frac{m}{m}$ identical particles do not interfere with each other. Then the absolute value of the drag force acting on a single particle is

$$\begin{aligned} |F| &= \frac{1}{\frac{m}{m}} \left| \int_{S_p} g[n_{sp}(\tilde{p} - p) - n_{sp} \cdot \tilde{\Pi}] dS \right| = \\ &= \frac{VG}{\frac{m}{m}} (1-\alpha) \rho |A_{\text{drag}}| \quad . \end{aligned} \quad (4.83)$$

In terms of the drag coefficient c_D , the force is (Schlichting, 1960, p. 15)

$$|F| = \frac{1}{2} c_D |u - u^*|^2 a_p \quad , \quad (4.84)$$

where a_p is the frontal area of the particle. Eliminating $|F|$ between Eqs. (4.83) and (4.84) one obtains

$$|A_{\text{drag}}| = \frac{1}{2} c_D |u - u^*|^2 a_p \frac{\frac{m}{m}}{VG} \frac{1}{1-\alpha} \quad , \quad (4.85)$$

or, using Eq. (4.12),

$$|A_{\text{drag}}| = \frac{1}{2} c_D |u-u^*|^2 \frac{\rho_p(d^*)}{v_p(d^*)} \quad (4.86)$$

The drag coefficient for a single sphere can be approximated by

$$c_D = 24/R_e + 0.4 \quad (4.87)$$

where

$$R_e = |u-u^*| \rho D_p(d^*)/u \quad (4.88)$$

is the particle Reynolds number and $D_p(d^*)$ is the average particle diameter. (About the approximation (4.87), see Figure 1.5 in Schlichting, 1960, p. 16.)

Substituting the expression (4.87) into (4.86) and observing that the acceleration is in the direction of $u-u^*$ one obtains for not interfering spheres the Reynolds formula

$$A_{\text{Reynolds}} = (u-u^*) \frac{\rho_p(d^*)}{v_p(d^*)} \left(0.2|u-u^*| + 12 \frac{\mu}{\rho D_p(d^*)} \right) \quad (4.89)$$

For a packed bed one finds, e.g., the Ergun correlation (Gibeling et al., 1980, pp. 15 and 30)

$$A_{\text{Ergun}} = (u-u^*) \frac{\rho_p(d^*)}{v_p(d^*)} \frac{2}{3} \frac{1}{\alpha^2} \left(1.75|u-u^*| + 150(1-\alpha) \frac{\mu}{\rho D_p(d^*)} \right) \quad (4.90)$$

In order to interpolate between both formulas one may assign limits for their validity. For instance, one could assume that the dispersed sphere formula holds for $\alpha > 0.9$, and the compacted sphere formula holds for $\alpha < 0.65$. Then the acceleration term is

$$A_{\text{drag}} = \begin{cases} A_{\text{Reynolds}} & \text{for } \alpha > 0.9 \\ 4[(\alpha-0.65)A_{\text{Reynolds}} + (0.9-\alpha)A_{\text{Ergun}}] & \text{for } 0.65 < \alpha < 0.9 \\ A_{\text{Ergun}} & \text{for } \alpha < 0.65 \end{cases} \quad (4.91)$$

The quoted limits are arbitrary and may be changed, if experiments are available. Also, other than Ergun formulas may be used, if experimental data indicate a better approach.

4.7.6 Acceleration by Granular Stresses. Acceleration by granular stresses enters the governing equations (4.32) for the particle velocity \dot{u} . The term is formally defined by (see Section 3.3)

$$A_{\text{stress}} = -\frac{1}{(1-\alpha)\rho} \nabla \cdot [(1-\alpha)\dot{\Pi}] + \frac{1}{(1-\alpha)\rho} \nabla \cdot [(1-\alpha)\dot{\Pi}_T] \quad (4.92)$$

The second term of Eq. (4.92) represents the acceleration of the particulate phase by solid phase turbulence which may be modeled by a solid phase turbulent stress tensor $\dot{\Pi}_T$. Because the density of the solid phase is much larger than that of the gas phase, and the sizes of the propellant grains are large, the turbulence of the solid phase is assumed negligible and $\dot{\Pi}_T$ is set equal to zero.

In the first term of Eq. (4.92), the variable $\dot{\Pi}$ models

$$\frac{1}{1-\alpha} \frac{1}{VG} \int_V (1-\beta) g (\dot{\Pi} + pI) dV \quad (4.93)$$

(see Eq. (3.44)). It is interpreted physically as the effect of grain interaction with grains. Without such an interaction, the stresses $\dot{\Pi}$ inside the grains would be equal to the negative of the surrounding gas pressure or nearly so, and the acceleration term A_{stress} could be neglected, except for turbulence considerations.

Generally in interior ballistics, one makes two assumptions about the model $\dot{\Pi}$ of the average intergranular stresses. First, one assumes that it is a function of α only and, second, one assumes that it is a diagonal matrix, i.e.,

$$\dot{\Pi} = I R_p(\alpha) \quad (4.94)$$

The second assumption means that the stresses have the effect of a pressure that acts on the particles in addition to the gas pressure. With these assumptions, one obtains from Eq. (4.92) for the acceleration

$$A_{\text{stress}} = -\frac{1}{1-\alpha} \frac{d}{d\alpha} \left[\frac{1-\alpha}{\rho} R_p(\alpha) \right] \nabla(1-\alpha) \quad (4.95)$$

The derivative term in Eq. (4.95) is interpreted as the square of the sound speed \dot{a} in the dispersed phase, and A_{stress} is expressed as

$$A_{\text{stress}} = - a^{*2}(\alpha) \frac{1}{1-\alpha} \nabla(1-\alpha) \quad (4.96)$$

The modeling of A_{stress} is reduced by these assumptions to the modeling of a sound speed function $a^*(\alpha)$. The sound speed can be measured in packed beds and in suspended particle flows, so that within a limited range the model can be tested.

The function $a^*(\alpha)$ should increase with higher particle density $(1-\alpha)\rho^*$, i.e., with decreasing α . Also, as α approaches one, the function should approach zero. Let a_{sp} be the sound speed within a particle and let us assume that for $\alpha = \alpha_1$ all particles touch each other, so that $a^*(\alpha_1) = a_{\text{sp}}$. Let $a^*(\alpha)$ become zero at $\alpha = \alpha_2 < 1$. Then a reasonable model for $a^*(\alpha)$ is

$$a^*(\alpha) = \begin{cases} a_{\text{sp}} \left(\frac{\alpha_1 - \alpha_0}{\alpha - \alpha_0} \right) \left(\frac{\alpha_2 - \alpha}{\alpha_2 - \alpha_1} \right) & \text{for } \alpha_0 < \alpha < \alpha_2, \\ 0, & \text{for } \alpha_2 < \alpha \end{cases} \quad (4.97)$$

In Eq. (4.97), the value $\alpha = \alpha_0$ corresponds to a highest density $(1-\alpha_0)\rho^*$ that can be achieved by compacting the particles. If $\alpha_0 = 0$ then one assumes that the particles can be crushed and compacted to a solid mass with the density ρ . The last factor in Eq. (4.97) merely lets a^* approach zero as α approaches α_2 . Thus, one assumes that for $\alpha > \alpha_2$ particle interaction can be neglected. Gibeling et al. (1980), uses a similar formula in which $\alpha_0 = 0$ and the second factor is set equal to one. Using that formula, one sets $a^*(\alpha) \equiv 0$ for $\alpha > \alpha_1$. It seems that a smooth transition to zero, as provided by our formula (4.97), is more realistic.

4.7.7 Burning Rate. The burning or regression rate directly enters the governing equation for the regression distance \tilde{d} in Eqs. (4.32). The corresponding term is defined as the surface average of the local regression rate $\partial \tilde{d} / \partial t = (\tilde{u}_{\text{sp}} - \tilde{u}) \cdot \mathbf{n}_{\text{sp}}$ (see Section 3.2.1) and is approximated by

$$\langle \dot{\tilde{d}} \rangle = \frac{1}{SG} \int_{S_p} g \frac{\partial \tilde{d}}{\partial t} ds \quad (4.98)$$

The linear regression rate can be measured, e.g., in closed bomb or strand burner experiments. The experiments show a dependence of the burning rate on the gas pressure, on gas velocity (erosive burning) and on the time derivative of the pressure (dynamic burning). Best established is the dependence of the burning rate on pressure, which is modeled by the equation

$$\dot{d}_s = B_0 + B_1 p^{B_2} \quad (4.99)$$

with constant B_0 , B_1 , and B_2 . The dependences on the relative velocity $|u - \dot{u}|^*$ and on the pressure change $\partial p / \partial t$ can be incorporated into the model equation either as additive terms or as a factor. The simplest model $\langle \dot{d} \rangle$ is obtained by neglecting these dependences and setting $\langle \dot{d} \rangle$ equal to \dot{d}_s , i.e.,

$$\langle \dot{d} \rangle = B_0 + B_1 p^{B_2} \quad (4.100)$$

The largest uncertainty of this model comes from the experimentally determined model parameters, and from the a priori assumptions that erosive and/or dynamic burning is, or is not important. An averaging error is also introduced by the use of the equation of state function $p(s, q)$ in Eq. (4.100). However, that error is likely to be negligible compared to the general inaccuracy of the model function. These errors are included in the error estimate (3.22).

4.7.8 Source Terms. In this section, we discuss terms in Eq. (4.32) that are associated with the burning of the propellant. They are characterized by the factor $\langle \dot{d} \rangle$, which represents the regression rate correlation and is discussed in Section 4.7.7. Because of this factor, the source terms are equal to zero if no burning takes place, and they represent sources of mass, energy, and momentum if the grains are burning. In the governing Eqs. (4.32), the terms have the common factor Γ and they enter the equations for s , q , and u . The factor Γ models (Section 3.3)

$$\frac{1}{\alpha} \frac{\rho}{\rho} \frac{1}{VG} \int_{S_p} g(\tilde{u}_{sp} - \dot{u}) \cdot \tilde{n}_{sp} dS \quad , \quad 1/s \quad , \quad (4.101)$$

and is defined by

$$\Gamma = \frac{1}{\alpha} \frac{\rho}{\rho} \frac{SG}{VG} \langle \dot{d} \rangle \quad (4.102)$$

In Eq. (4.102), SG can be eliminated using the relation (4.19). The result is

$$\Gamma = \frac{1}{\alpha} \frac{\rho}{\rho} \frac{m}{VG} s_p(\dot{d}) \langle \dot{d} \rangle \quad , \quad (4.103)$$

as stated by Eq. (4.33).

The approximation error in Eq. (4.102) is that of the correlation $\langle \tilde{d} \rangle$ (see Section 3.2.1). In the expression (4.103) one has, in addition, errors associated with the representation of the weighted surface SG by the product \tilde{m}_p . Because the representation is part of the definition of \tilde{m} (see Section 4.2.1), it does not formally introduce new errors.

The governing equation (4.32) for the gas velocity contains the source term $(\tilde{u}-u)\Gamma$. The term models

$$\frac{1}{\alpha} \frac{\rho}{\rho} \frac{1}{VG} \int_{S_p} (\tilde{u}-u) g [(\tilde{u}_{sp}-\tilde{u}) \cdot n_{sp}] dS, \quad \text{m/s}^2. \quad (4.104)$$

The error in the governing equation caused by the model (4.103) is

$$\frac{1}{\alpha} \frac{\rho}{\rho} \frac{1}{VG} \int_{S_p} (\tilde{u}-u) g [(\tilde{u}_{sp}-\tilde{u}) \cdot n_{sp}] dS. \quad (4.105)$$

The error is zero if the grains do not rotate and all grains have the same velocity.

The entropy governing equation (4.32) contains the source term $H\Gamma$. The term is derived under the assumption that the approximation

$$\int_{S_p} g \tilde{e} (\tilde{u}_{sp}-\tilde{u}) \cdot n_{sp} dS = \hat{e} \int_{S_p} g (\tilde{u}_{sp}-\tilde{u}) \cdot n_{sp} dS \quad (4.106)$$

holds. Eq. (4.106) is indeed an identity if the local specific energy \tilde{e} of the gas released from the burning propellant surface is equal to a constant \hat{e} . This is a common assumption in interior ballistics. The constant \hat{e} is the specific energy of the gas at "flame temperature", i.e.,

$$\hat{e} = \frac{1}{\gamma-1} \frac{R}{M} T_{\text{flame}} = \frac{1}{\gamma-1} g_a I_p, \quad \text{J/kg}, \quad (4.107)$$

where g_a is the standard acceleration 9.80665 m/s^2 and

$$I_p = T_{\text{flame}} R/(g_a M), \quad \text{m}, \quad (4.108)$$

is the "force" or "impetus" of the propellant. (Sometimes also the product $g_a I_p$ (m^2/s^2) is called the "impetus" of the propellant.)

In some cases, a modeling of \hat{e} may be better than the assumption of a constant \hat{e} . For instance, if the propellant contains a retardant then one could assume that the flame temperature is a function of the regression distance and, consequently, $\hat{e} = \hat{e}(\dot{d})$. Of course, the modeling then involves averaging errors, because the local $\hat{e}(\dot{d})$ would be replaced by a function $\hat{e}(\dot{d})$ of the average \dot{d} .

The factor H is defined by

$$H = \frac{1}{T} [(\hat{e} + p/\rho) - (e + p/\rho)] \quad , \quad \text{J/(kg}\cdot\text{K)} \quad , \quad (4.109)$$

i.e., H is the difference between the enthalpy of the gas emerging from the flame and of the surrounding gas, divided by the gas temperature. The approximations that affect this term are those of the equations of state (see Section 4.7.1).

The source term in the governing equation, Eq. (4.32), for the pressure logarithm function q has as a factor of Γ the expression $(\hat{e} - e - e_s H)/e_q$, where, $e_s(s, q)$ and $e_q(s, q)$ are the partial derivatives of the specific internal energy e with respect to s and q, respectively. The factor is derived by formal manipulation and approximations involved in the derivation are the same as discussed above.

4.7.9. Grain Volume and Surface. We recall the discussions in Section 4.2.1 about the definition of the grain number function \dot{m} . The formal definition of the average grain volume function $v_p(\dot{d})$ and of the average grain surface function $s_p(\dot{d})$ should be consistent with the definition of \dot{m} . In this section, we shall discuss definitions that are consistent with Eqs. (4.18) and (4.19), respectively.

For convenience, we repeat the pertinent equations and definitions in this section. Our goal is to find such functions \dot{m} , v_p , and s_p that satisfy the following set of relations

$$\dot{d}(t, x) = \frac{1}{SG} \int_{S_p} g \dot{d} \, dS \quad . \quad (4.110)$$

$$\frac{dv_p(\dot{d})}{d\dot{d}} = -s_p(\dot{d}) \quad , \quad (4.111)$$

$$\dot{m}(t, x) s_p(\dot{d}) = \int_{S_p} g \, dS \quad , \quad (4.112)$$

$$\dot{m}(t, x) v_p(\dot{d}) = \int_V (1-\beta) g \, dV \quad . \quad (4.113)$$

We found in Section 4.2.1, that such functions in general do not exist and, therefore, suggested to define \bar{m}^* by either of the following two equations:

$$\bar{m}^* = \sum_{i=1}^n \left\{ \frac{1}{s_{pi}} \int_{s_i \cap V} g \, dS \right\} , \quad (4.114)$$

or

$$\bar{m}^* = \sum_{i=1}^n \left\{ \frac{1}{v_{pi}} \int_{v_i \cap V} g \, dV \right\} . \quad (4.115)$$

Once \bar{m}^* is defined, then one can define either s_p or v_p by Eqs. (4.112) or (4.113), respectively, and find the other function from Eq. (4.111).

The approximations involved are, first, due to the assumption that \bar{m}^* , as defined, is independent of \bar{d} . The accuracy of the approximation is improved if the weight function g is constant over most of the averaging volume. A second approximation is due to the assumption that s_p or v_p , defined by Eqs. (4.112) or (4.113), respectively do not depend explicitly on t and x . Again, an almost constant g may improve the accuracy of this approximation.

The modeling of the functions v_p and s_p practically is done at a limit, assuming constant g , and identical particles. In this case, the functions simply represent a single particle.

If there is a variation of particle sizes within the averaging volume, then by either of the described formalisms one obtains an average that is slanted towards the larger particles. Investigations of the significance of this bias have not been done for interior ballistics problems.

4.7.10. Grain Surface Heating Rate. The grain surface heating rate enters the governing equation, Eq. (4.32), for the grain surface temperature

$$\frac{\partial \bar{T}}{\partial t} = - \bar{u} \cdot \nabla \bar{T} + \langle \dot{T} \rangle . \quad (4.116)$$

The term $\langle \dot{T} \rangle$ is the correlation model for

$$\dot{T} = \frac{1}{SG} \int_{s_p} \frac{\partial \bar{T}}{\partial t} g \, dS , \quad (4.117)$$

i.e., for the average rate of change of the surface temperature. The change is related to the heat flux to the particles, \dot{e} , discussed in Section 4.7.4. Therefore, the model $\langle \dot{T} \rangle$ should be consistent with the model $\langle \dot{e} \rangle$.

Like the grain surface and grain volume functions, the surface temperature model function is usually established by considering the limiting case of identical grains, i.e., by treating a single grain. Typically, if the grain has a simple geometry, one calculates the temperature field within the grain corresponding to the energy transfer $\langle \dot{e} \rangle$. This involves the solution of a differential equation and finding the corresponding surface temperature, which, in turn, determines the energy transfer at the next time step. This type of calculation is recommended if one is particularly interested in the ignition process. After ignition, all heat transfer is assumed to be zero, because then the energy flow phenomena are dominated by the combustion and the associated heat release. The continued heating of the grains is assumed to be of no consequence for the combustion.

In order to illustrate the relation between the heat transfer for the gas to the particles, (4.77), and $\langle \dot{T} \rangle$, we consider a very simple model in which the temperature in each grain is assumed to be uniform. (The model is not recommended for simulation of interior ballistics, but it shows the salient features of the relation.) Let \dot{c}_p be the specific heat of the particle material. Then the heat capacity of one particle is $\dot{c}_p \dot{v}_p$, (J/K). Therefore, the relation between the energy transfer models $\langle \dot{e} \rangle$ and $\langle \dot{T} \rangle$ should be

$$\dot{m} \dot{c}_p \dot{v}_p \langle \dot{T} \rangle = \langle \dot{e} \rangle SG \quad (4.118)$$

From Eq. (4.118) and expression (4.77), the model for the heat conduction between the gas and particles can be written in terms of $\langle \dot{T} \rangle$ as

$$\dot{v}_{\text{particle}} = - \frac{1}{\alpha \rho T} \frac{\dot{m}}{VG} [\dot{c}_p \dot{v}_p \langle \dot{T} \rangle] \quad (4.119)$$

The important result is the existence of a relation like Eq. (4.118) between $\langle \dot{T} \rangle$ and $\langle \dot{e} \rangle$. It would be replaced by a different relation if the heat flow within the particle were taken into account, as described above. In that case, the expression in the brackets in Eq. (4.119) would be changed correspondingly.

5. SUMMARY AND CONCLUSIONS

Interior ballistics models are mostly based on engineering approximations and insight, like Lagrange's model. Alternatively, one can assume that the gas and particles locally satisfy all conservation equations and obtain the model by an averaging process. In this report, we present a complete mathematical derivation of weighted volume averaged equations including all error terms, sufficient conditions for the necessary differentiability of the average variables, and regions of definition of the average variables. Initial and boundary conditions that are consistent with the volume averaged equations are discussed. Correlations that are used to close the system of partial differential equations are examined. Some of these correlations are different than those commonly used in interior ballistic applications.

The average governing equations that are derived in this report model the transient effects of viscosity, heat conduction, and turbulence in the compressible gas phase; the ignition, intergranular stress, and burning in the incompressible solid phase; and the corresponding interactions between the phases, e.g., drag, heat transfer, and source terms. Turbulence is defined in terms of volume averages and only elementary models are presented for completeness of the report. In the average model, quantities appear that are defined only on the surfaces of the grains. We show that these quantities satisfy a general partial differential equation. The relationships between the volume average equations and the local equations for individual phases are discussed as the volume of the solid phase approaches zero and as the size of the averaging volume approaches zero. Because these equations must be solved via the computer, an appropriate form and choice of dependent variables for numerical solution are discussed. Thus, this report presents a complete and consistent mathematical model of interior ballistics for non-reacting, gas-solid flows.

The exposition of the theoretical basis of averaged equations permits us to draw the following conclusions:

First, the proper domain for averaging is a volume that is larger than the propellant grains and that is smaller than the gun tube. Time averaging is undesirable because of the rapid changes of the flow field and the moving boundary (projectile). Infinite volume averaging is not admissible for theoretical reasons, and so are surface and line averages.

Second, the average equations are valid only for cases where the averaging volume consists of gas and particles or just gas and where the local functions have no discontinuities within their respective domains. Therefore, average governing equations are not suitable for describing flows with shocks, contact discontinuities, etc. On the other hand, by a proper formulation of the governing equations, we obtain a system that can be solved numerically without explicitly following the boundaries of regions without particles.

Third, the average equations are not valid in boundary regions. Consequently, the formulation of proper boundary conditions is problematic, and has not been solved satisfactorily. Also, resolution of interior ballistics boundary layers based on volume average two-phase equations is only possible in exceptional cases, when the grains are smaller than the typical boundary layer.

Fourth, one-dimensional interior ballistics models based on volume averaging are less problematic than two-dimensional models, because the averaging volume occupies a finite thickness cross-section of the tube and is large compared to the particles. The only problems with such models are the formulation of boundary conditions at the breech and projectile.

Fifth, a mathematical basis for two-dimensional interior ballistic models could possibly be obtained by an extension of the theory of average equations. Such an extension can be done by generalizing to a variable volume averaging or by using statistical averages instead of volume averages. The first approach will alleviate some problems, but it cannot remove the basic cause of problems in two-dimensional modeling: the particle sizes that are large compared to the gas boundary layer. The second approach (statistical averaging) has not been tried successfully for two-phase flows. There the encountered problems are mathematical, requiring a major investment in the development of the theory.

LIST OF SYMBOLS

The list contains symbols that are frequently used in the report. Symbols that are defined and used only locally are not included in this list.

Function symbols in general indicate average quantities. A tilde over a function symbol is used to indicate the local value of a function. An asterisk over a symbol indicates that it represents a property of the propellant grains.

a	- sound speed in gas, m/s
a_{sp}	- sound speed of particle material, m/s
$\overset{*}{a}$	- sound speed of particulate phase, m/s
a_p	- average frontal area of a particle, m^2
A_{drag}	- acceleration term due to drag, m/s^2
A_{Ergun}	- Ergun correlation for A_{drag} , m/s^2
$A_{Reynolds}$	- Reynolds correlation for A_{drag} , m/s^2
A_{stress}	- acceleration term due to intergranular stress, m/s^2
c_v	- specific heat capacity at constant volume, J/(kg·K)
c_p	- specific heat capacity at constant pressure, J/(kg·K)
$\overset{*}{d}$	- regression distance, m
\dot{d}_s	- stationary burning rate, m/s
$\langle \dot{d} \rangle$	- burning rate correlation function, m/s
\bar{d}_p	- average particle diameter, m
e	- specific internal energy, J/kg
\bar{e}	- e at flame temperature, J/kg
e_s, e_q	- partial derivatives of e , K and m^3/kg
$\langle \dot{e} \rangle$	- correlation for surface averaged heat flux into the particles, W/m^2
E	- strain rate tensor, 1/s
g	- averaging weight function
H	- specific enthalpy difference (Section 4.7.8), J/(kg·K)
I	- identity tensor of second order

l	- diameter of averaging volume, m
n	- number of grains in averaging volume
$\frac{\star}{n}$	- weighted number of grains in averaging volume
M	- molar mass, kg/mol
n_{sp}	- unit outward normal with respect to the gas on S_p
n_{sv}	- unit outward normal to S_v
p	- pressure, Pa
p_q	- derivative of the function $p(q)$
Q	- gas phase heat conduction, W/m^2
Q_T	- gas phase turbulent heat flux, W/m^2
q	- pressure logarithm function (4.22), Pa
r	- radial coordinate, m
$R=8.3143 \text{ J/(mol} \cdot \text{K)}$	- universal gas constant
s	- specific entropy, $J/(kg \cdot K)$
s_p	- average surface area of a single grain, m^2
S_p	- union of all grain surfaces in V
SG	- weighted area of S_p , m^2
S_v	- surface of averaging volume V
t	- time, s
T	- gas temperature, K
T_{flame}	- flame temperature, K
$\frac{\star}{T}$	- grain surface temperature, K
$\langle \dot{T} \rangle$	- correlation for rate of change of grain surface temperature, K/s
u	- gas velocity, m/s
u_r, u_θ, u_z	- the radial, circumferential, and axial components of u , m/s
$\frac{\star}{u}$	- particle velocity, m/s

u_r^*, u_θ^*, u_z^*	- the radial, circumferential, and axial components of \vec{u} , m/s
\vec{u}_{sp}	- velocity of a point of S_p , m/s
v_p	- average value of the volume of a single particle, m^3
V	- averaging volume
VG	- weighted value of V , m^3
x	- spacial coordinate, m
z	- axial coordinate, m
Z	- surface element metric
α	- gas volume fraction (porosity)
β	- phasic function (Section 2)
γ	- ratio of specific heats
Γ	- source term, (4.33) 1/s
Γ_1	- $SG\langle\dot{d}\rangle/VG$, 1/s
Γ_2	- $\frac{\alpha\rho}{\rho^*} \Gamma$, 1/s
η	- covolume in equation of state, m^3/kg
κ	- thermal conductivity coefficient, $W/(m\cdot K)$
λ	- bulk viscosity coefficient, $Pa\cdot s$
μ	- shear viscosity coefficient, $Pa\cdot s$
Π	- viscous stress tensor, Pa
Π_T	- gas phase turbulent stress tensor, Pa
Π^*	- intergranular stress tensor, Pa
Π_T^*	- solid phase turbulent stress tensor, Pa
ρ	- gas density, kg/m^3
ρ^*	- particle density, kg/m^3
ρ_s, ρ_q	- partial derivatives of $\rho(s, q)$, (kg/m^3) $(kg\cdot K/J)$, and s^2/m^2
ϕ	- function describing a gas property

ϕ	- function describing a particle property
ϕ	- dissipation term, $W/(kg \cdot K)$
$\bar{\phi} = \phi_L$	- dissipation function, \dot{W}/m^3
ϕ_T	- gas phase turbulent dissipation function, W/m^3
$\langle \phi \rangle$	- dissipation correlation term, $W/(kg \cdot K)$
ϕ_1	- $\rho T \phi$, W/m^3
$\psi(t, x, \xi)$	- general function, Section 2
γ	- heat conduction term, $W/(kg \cdot K)$
γ_{gas}	- heat conduction due to gas conductivity, $W/(kg \cdot K)$
$\gamma_{particle}$	- heat conduction due to heat loss to particles, $W/(kg \cdot K)$

REFERENCES

1. T. Apostol, Mathematical Analysis, 1st Ed., Addison-Wesley Publishing Co., Inc., New York, 1957.
2. T. Cebeci and A. Smith, Analysis of Turbulent Boundary Layers, Academic Press, New York, 1974.
3. R. Courant and F. John, Introduction to Calculus and Analysis, Vol. II, pp. 459-462, John Wiley and Sons, Inc., New York, 1974.
4. J.M. Delhay and J.L. Achard, "On the Use of Averaging Operators in Two-Phase Modeling," in Thermal and Hydraulic Aspects of Nuclear Reactor Safety, Vol. 1: Light Water Reactors, O.C. Jones and S.G. Bankoff, eds., pp. 289-332, ASME, New York, 1977.
5. E.B. Fisher and A.P. Trippe, "Mathematical Model of Center Core Ignition in the 175mm Gun," Calspan Report VQ-5163-D-2, 1974.
6. W. Fulks, Advanced Calculus, 2nd Ed., John Wiley and Sons, Inc., New York, 1969.
7. H.J. Gibeling, R.C. Buggeln, and H. McDonald, "Development of a Two-Dimensional Implicit Interior Ballistics Code," USA ARRADCOM/Ballistic Research Laboratory Contractor Report, ARBRL-CR-00411, APG, MD, January 1980.
8. P.S. Gough, "The Flow of a Compressible Gas Through an Aggregate of Mobile, Reacting Particles," Ph.D. Thesis, Department of Mechanical Engineering, McGill University, Montreal, 1974.
9. F. Hund, Einführung in die Theoretische Physik, Bd. 4, "Theorie der Wärme," p. 135 ff, Bibliographisches Institut, Leipzig, 1950.
10. M. Ishii, Thermo-Fluid Dynamic Theory of Two-Phase Flow, Eyrolles, France, 1975.
11. H. Krier, W.F. van Tassell, S. Rajan, and J. Vershaw, "Model of Flamespreading and Combustion Through Packed Beds of Propellant Grains," University of Illinois at Urbana-Champaign Report, TR-AAE-74-1, 1974.
12. K.K. Kuo, J.H. Koo, T.R. Davis, and G.R. Coates, "Transient Combustion in Mobile, Gas-Permeable Propellants," Acta. Astron., Vol. 3, No. 7-8, pp. 574-591, 1976.
13. W. Prager, Introduction to Mechanics of Continua, Ginn and Company, New York, 1961.
14. H. Schlichting, Boundary Layer Theory, 4th Ed., McGraw-Hill Book Company, New York, 1960.

15. C. Truesdell and R. Toupin, "The Classical Field Theories," in Encyclopedia of Physics, S. Flügge, ed., Vol. III/1, Springer-Verlag, 1960.
16. H.S. Tsien, "The Equation of Gas Dynamics," in Fundamentals of Gas Dynamics, H.W. Emmons, ed., Princeton University Press, Princeton, NJ, 1958.

Appendix A

Governing Equations for Cylindrically Symmetric Flows in Cylindrical Coordinates.

This appendix contains a list of the governing equations in component form in cylindrical coordinates for the case of cylindrical symmetry flow. The subscripted variables denote the components of a vector and not the derivative of these variables. All derivatives are written in a non-abbreviated form. The listed equations are in a form which is compatible with Eq. (4.32). The components of the gas average velocity and the particle average velocity are

$$u = (u_r, u_\theta, u_z), \quad (\text{m/s}), \quad (\text{A.1})$$

$$\dot{u} = (\dot{u}_r, \dot{u}_\theta, \dot{u}_z), \quad (\text{m/s}), \quad (\text{A.2})$$

where the subscripts r , θ , and z refer to the radial, angular, and axial coordinate directions, respectively. The components of the gradient of a scalar f are

$$\nabla f = \left\{ \left(\frac{\partial f}{\partial r} \right)_r, (0)_\theta, \left(\frac{\partial f}{\partial z} \right)_z \right\}. \quad (\text{A.3})$$

The divergence of a vector $F = (F_r, F_\theta, F_z)$ is

$$\nabla \cdot F = \frac{1}{r} \frac{\partial(rF_r)}{\partial r} + \frac{\partial F_z}{\partial z}. \quad (\text{A.4})$$

The independent variables are time t , radial position r , and axial position z . The dependent average variables which are computed from the governing partial differential equations are: the specific entropy s , the pressure logarithm function q , the radial gas velocity u_r , the circumferential gas velocity u_θ , the axial gas velocity u_z , the radial particle velocity \dot{u}_r , the circumferential particle velocity \dot{u}_θ , the axial particle velocity \dot{u}_z , the number of particles within the averaging volume \dot{m} , the regression distance \dot{d} , and the surface temperature of the particles \dot{T} .

The entropy equation is

$$\frac{\partial s}{\partial t} = -u_r \frac{\partial s}{\partial r} - u_z \frac{\partial s}{\partial z} + \frac{p}{\rho T} B + H \Gamma + \Phi + \Psi \quad (\text{A.5})$$

where p , ρ , T , H , Γ are given by Eqs. (B.6), (B.4), (B.2), (B.27), and (B.26), respectively. The expression for B is

$$B = -\frac{1}{\alpha} \left\{ (1-\alpha) \left[\frac{1}{r} \frac{\partial(r u_r^*)}{\partial r} + \frac{\partial u_z^*}{\partial z} \right] - (u_r^* - u_r) \frac{\partial(1-\alpha)}{\partial r} - (u_z^* - u_z) \frac{\partial(1-\alpha)}{\partial z} \right\}, \quad (A.6)$$

and the porosity α is given by Eq. (B.1). The dissipation function ϕ is

$$\phi = \frac{1}{\rho T} \bar{\phi}(E) + \frac{1}{\rho T} \phi_T + \langle \phi \rangle, \quad (A.7)$$

where

$$\begin{aligned} \bar{\phi}(E) = & \frac{4}{3} \mu \left[\left(\frac{\partial u_r}{\partial r} \right)^2 + \left(\frac{u_r}{r} \right)^2 + \left(\frac{\partial u_z}{\partial z} \right)^2 - \left(\frac{u_r}{r} \frac{\partial u_r}{\partial r} + \frac{\partial u_r}{\partial r} \frac{\partial u_z}{\partial z} + \frac{u_r}{r} \frac{\partial u_z}{\partial z} \right) \right] \\ & + \mu \left[\left(r \frac{\partial}{\partial r} \left(\frac{u_\theta}{r} \right) \right)^2 + \left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right)^2 + \left(\frac{\partial u_\theta}{\partial z} \right)^2 \right] \\ & + \lambda \left[\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} \right]^2, \end{aligned} \quad (A.8)$$

and μ , λ , $\langle \phi \rangle$, and ϕ_T are given by Eqs. (B.7), (B.8), (B.13), and (B.35), respectively. The heat conduction term Ψ is given by Eq. (B.15) as

$$\Psi = \Psi_{\text{gas}} + \Psi_{\text{particle}} + \Psi_{\text{turb}}, \quad (A.9)$$

where

$$\Psi_{\text{gas}} = \frac{1}{\alpha \rho T} \left[\frac{1}{r} \frac{\partial}{\partial r} (r \alpha \kappa \frac{\partial T}{\partial r}) + \frac{\partial}{\partial z} (\alpha \kappa \frac{\partial T}{\partial z}) \right], \quad (A.10)$$

$$\begin{aligned} \Psi_{\text{turb}} = & \frac{1}{\alpha \rho T} \left[\frac{1}{r} \frac{\partial}{\partial r} (r \alpha \kappa_T \frac{\partial T}{\partial r}) + \frac{\partial}{\partial z} (\alpha \kappa_T \frac{\partial T}{\partial z}) \right. \\ & \left. - \frac{1}{r} \frac{\partial}{\partial r} (r \kappa_T (T_i - T) \frac{\partial \alpha}{\partial r}) - \frac{\partial}{\partial z} (\kappa_T (T_i - T) \frac{\partial \alpha}{\partial z}) \right], \end{aligned} \quad (A.11)$$

and Ψ_{particle} , κ are given by Eqs. (B.17), (B.14), respectively, and κ_T, T_i are discussed near Eq. (B.36).

The pressure logarithm function equation is

$$\begin{aligned} \frac{\partial q}{\partial t} = & -u_r \frac{\partial q}{\partial r} - u_z \frac{\partial q}{\partial z} - \frac{\rho}{\partial q} \left(\frac{1}{r} \frac{\partial(ru_r)}{\partial r} + \frac{\partial u_z}{\partial z} + \frac{\partial e}{\partial s} \frac{1}{T} B \right) \\ & + \frac{1}{\frac{\partial e}{\partial q}} \left(\hat{e} - e - \frac{\partial e}{\partial s} H \right) \Gamma + \frac{\partial \rho}{\partial s} \frac{1}{\frac{\partial \rho}{\partial q}} (\phi + \psi) , \end{aligned} \quad (A.12)$$

where ρ , e , T , B , \hat{e} , H , Γ , ϕ , and ψ are given by Eqs. (B.4), (B.3), (B.2), (A.6), (B.28), (B.27), (B.26), (A.7), and (A.9), respectively.

The radial gas velocity equation is

$$\begin{aligned} \frac{\partial u_r}{\partial t} = & -u_r \frac{\partial u_r}{\partial r} - u_z \frac{\partial u_r}{\partial z} + \frac{u_\theta^2}{r} - \frac{dp}{dq} \frac{1}{\rho} \frac{\partial q}{\partial r} - (u_r - u_r^*) \Gamma \\ & - \frac{(1-\alpha)}{\alpha} (A_{\text{drag}})_r + (A_{\text{visc}})_r + (A_{\text{turb}})_r , \end{aligned} \quad (A.13)$$

where

$$\begin{aligned} (A_{\text{visc}})_r = & \frac{1}{\alpha \rho} \left\{ \frac{\partial}{\partial r} \left[\alpha \mu \frac{2}{3} \left(2 \frac{\partial u_r}{\partial r} - \frac{u_r}{r} - \frac{\partial u_z}{\partial z} \right) + \alpha \lambda \left(\frac{1}{r} \frac{\partial(ru_r)}{\partial r} + \frac{\partial u_z}{\partial z} \right) \right] \right. \\ & \left. + \frac{\partial}{\partial z} \left[\alpha \mu \left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right] + 2\alpha \mu \frac{\partial}{\partial r} \left(\frac{u_r}{r} \right) \right\} , \end{aligned} \quad (A.14)$$

and p , ρ , Γ , α , μ , and λ are given by Eqs. (B.6), (B.4), (B.26), (B.1), (B.7), and (B.8), respectively. The radial component of the drag $(A_{\text{drag}})_r$ is given by the radial component of Eq. (B.20). The radial component of acceleration due to turbulence $(A_{\text{turb}})_r$ could be given by the radial component of Eq. (B.34) which is Eq. (A.14) with μ and λ replaced by μ_T and λ_T .

The circumferential gas velocity equation is

$$\begin{aligned} \frac{\partial u_\theta}{\partial t} = & -u_r \frac{\partial u_\theta}{\partial r} - u_z \frac{\partial u_\theta}{\partial z} - \frac{u_r u_\theta}{r} - (u_\theta - u_\theta^*) \Gamma \\ & - \frac{(1-\alpha)}{\alpha} (A_{\text{drag}})_\theta + (A_{\text{visc}})_\theta + (A_{\text{turb}})_\theta , \end{aligned} \quad (A.15)$$

where

$$(A_{\text{visc}})_\theta = \frac{1}{\alpha\rho} \left\{ \frac{\partial}{\partial r} \left[\alpha\mu r \frac{\partial}{\partial r} \left(\frac{u_\theta}{r} \right) \right] + \frac{\partial}{\partial z} \left[\alpha\mu \frac{\partial u_\theta}{\partial z} \right] + 2\alpha\mu \frac{\partial}{\partial r} \left(\frac{u_\theta}{r} \right) \right\} , \quad (\text{A.16})$$

and α , Γ , μ , λ , and ρ are given by Eqs. (B.1), (B.26), (B.7), (B.8), and (B.4), respectively. The circumferential component of the drag $(A_{\text{drag}})_\theta$ is given by the circumferential component of Eq. (B.20). The circumferential component of the acceleration due to turbulence $(A_{\text{turb}})_\theta$ could be given by the circumferential component of Eq. (B.34) which is Eq. (A.16) with μ and λ replaced by μ_T and λ_T .

The axial gas velocity equation is

$$\begin{aligned} \frac{\partial u_z}{\partial t} = & -u_r \frac{\partial u_z}{\partial r} - u_z \frac{\partial u_z}{\partial z} - \frac{dp}{dq} \frac{1}{\rho} \frac{\partial q}{\partial z} - (u_z - u_z^*) \Gamma \\ & - \frac{(1-\alpha)}{\alpha} (A_{\text{drag}})_z + (A_{\text{visc}})_z + (A_{\text{turb}})_z , \end{aligned} \quad (\text{A.17})$$

where

$$\begin{aligned} (A_{\text{visc}})_z = & \frac{1}{\alpha\rho} \left\{ \frac{\partial}{\partial r} \left[\alpha\mu \left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right] + \frac{\alpha\mu}{r} \left[\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right] \right. \\ & \left. + \frac{\partial}{\partial z} \left[\alpha\mu \frac{2}{3} \left(2 \frac{\partial u_z}{\partial z} - \frac{\partial u_r}{\partial r} - \frac{u_r}{r} \right) + \alpha\lambda \left(\frac{\partial u_z}{\partial z} + \frac{\partial u_r}{\partial r} + \frac{u_r}{r} \right) \right] \right\} , \end{aligned} \quad (\text{A.18})$$

and p , ρ , Γ , α , μ , and λ are given by Eqs. (B.6), (B.4), (B.26), (B.1), (B.7), and (B.8), respectively. The axial component of the drag $(A_{\text{drag}})_z$ is given by the axial component of Eq. (B.20). The axial component of the acceleration due to turbulence $(A_{\text{turb}})_z$ could be given by the axial component of Eq. (B.34) which is Eq. (A.18) with μ and λ replaced by μ_T and λ_T .

The components of the solid phase velocity equation are

the radial solid phase velocity equation

$$\frac{\partial u_r^*}{\partial t} = -u_r^* \frac{\partial u_r^*}{\partial r} - u_z^* \frac{\partial u_r^*}{\partial z} + \frac{u_\theta^{*2}}{r} - \frac{dp}{dq} \frac{1}{\rho^*} \frac{\partial q}{\partial r} + \frac{\rho}{\rho^*} (A_{\text{drag}})_r + (A_{\text{stress}})_r , \quad (\text{A.19})$$

the circumferential solid phase velocity equation

$$\frac{\partial u_{\theta}^*}{\partial t} = -u_r^* \frac{\partial u_{\theta}^*}{\partial r} - u_z^* \frac{\partial u_{\theta}^*}{\partial z} - \frac{u_r^* u_{\theta}^*}{r} + \frac{\rho}{\rho} (A_{\text{drag}})_{\theta} \quad , \quad (\text{A.20})$$

and the axial solid phase velocity equation

$$\frac{\partial u_z^*}{\partial t} = -u_r^* \frac{\partial u_z^*}{\partial r} - u_z^* \frac{\partial u_z^*}{\partial z} - \frac{dp}{dq} \frac{1}{\rho} \frac{\partial q}{\partial z} + \frac{\rho}{\rho} (A_{\text{drag}})_z + (A_{\text{stress}})_z \quad , \quad (\text{A.21})$$

where p and ρ are given by Eqs. (B.6) and (B.4), respectively. The density of the solid phase ρ is assumed constant. The components of the accelerations due to drag, A_{drag} , and intergranular stress, A_{stress} , are given by the components of Eqs. (B.20) and (B.23), respectively.

The particle number equation is

$$\frac{\partial m^*}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} (r m u_r^*) - \frac{\partial}{\partial z} (m u_z^*) \quad . \quad (\text{A.22})$$

The regression rate equation is

$$\frac{\partial d^*}{\partial t} = -u_r^* \frac{\partial d^*}{\partial r} - u_z^* \frac{\partial d^*}{\partial z} + \langle \dot{d} \rangle \quad , \quad (\text{A.23})$$

where the burning rate correlation $\langle \dot{d} \rangle$ is given by Eq. (B.25).

The surface temperature equation is

$$\frac{\partial T^*}{\partial t} = -u_r^* \frac{\partial T^*}{\partial r} - u_z^* \frac{\partial T^*}{\partial z} + \langle \dot{T} \rangle \quad , \quad (\text{A.24})$$

where the correlation $\langle \dot{T} \rangle$ for the rate of change of grain surface temperature is discussed in Section (4.7.10).

Appendix B

Correlation Model Formulas

This appendix contains a list of correlation model formulas. The formulas are discussed in detail in Section 4.7. The terms listed in this appendix are in a form compatible with Eq. (4.32) and those listed in Appendix A.

The porosity or gas volume fraction (Section 4.2.1) is given by

$$\alpha = 1 - v_p^* (d_m^*) / VG \quad . \quad (B.1)$$

The equations of state (Section 4.7.1) are

$$T(p,s) = T_R \left(\frac{p}{p_R} \right)^{(\gamma-1)/\gamma} \exp \left(\frac{M}{R} \frac{\gamma-1}{\gamma} s \right) , \quad K , \quad (B.2)$$

$$e = \frac{1}{\gamma-1} \frac{R}{M} T , \quad J/kg , \quad (B.3)$$

$$\rho = \left(\frac{R}{M} \frac{T}{p} + \eta \right)^{-1} , \quad kg/m^3 , \quad (B.4)$$

$$a^2 = \gamma \frac{p}{\rho} \frac{1}{1-\eta\rho} , \quad m^2/s^2 , \quad (B.5)$$

where $R = 8.3143 \text{ J/(mol}\cdot\text{K)}$ is the universal gas constant, $M \text{ (kg/mol)}$ is the molar mass and $\eta \text{ (m}^3\text{/kg)}$ is the covolume. The pressure logarithm function q is defined by (Section 4.2.2)

$$q = q_1 [\ln(p/p_1) + 1] , \quad Pa; \text{ or } p = p_1 \exp\left(\frac{q}{q_1} - 1\right) , \quad Pa . \quad (B.6)$$

The shear viscosity coefficient μ and the bulk viscosity coefficient λ are (Section 4.7.2)

$$\mu = \mu_0 + \mu_1 \frac{T^{1.5}}{\mu_2 + T} , \quad Pa\cdot s , \quad (B.7)$$

$$\lambda = \lambda_0 + \lambda_1 \frac{T^{1.5}}{\lambda_2 + T} , \quad Pa\cdot s . \quad (B.8)$$

The acceleration by viscosity is modeled by (Section 4.7.2)

$$A_{\text{visc}} = \frac{1}{\alpha \rho} \nabla \cdot \{ \alpha [2\mu E + (\lambda - \frac{2}{3} \mu) (\text{trace } E) I] \} , \quad \text{m/s}^2 \quad (\text{B.9})$$

where E is the strain rate tensor computed using the average velocities, i.e.,

$$E = 0.5 (\nabla u + (\nabla u)^T) . \quad (\text{B.10})$$

The heat dissipation function term is modeled by (Section 4.7.3)

$$\phi = \frac{1}{\rho T} \bar{\phi}(E) + \langle \phi \rangle + \frac{1}{\rho T} \phi_T , \quad \text{W/(kg}\cdot\text{K)} , \quad (\text{B.11})$$

where

$$\bar{\phi}(E) = 2\mu \text{trace}(E^2) + (\lambda - \frac{2}{3} \mu) (\text{trace } E)^2 , \quad \text{W/m}^3 , \quad (\text{B.12})$$

$$\langle \phi \rangle = \frac{1}{\rho T} |u - \bar{u}|^2 \left(\frac{\bar{m}}{4 \cdot VG} \right)^{2/3} \pi^2 \left(\frac{5}{3} \mu + \frac{1}{2} \lambda \right) , \quad \text{W/(kg}\cdot\text{K)} , \quad (\text{B.13})$$

and ϕ_T is given by Eq. (B.35).

The thermal conductivity coefficient κ is modeled by (Section 4.7.4)

$$\kappa = \kappa_0 + \kappa_1 \frac{T^{1.5}}{\kappa_2 + T} , \quad \text{W/(m}\cdot\text{K)} . \quad (\text{B.14})$$

The heat conduction term in the governing equations is modeled by (Section 4.7.4)

$$\psi = \psi_{\text{gas}} + \psi_{\text{particle}} + \psi_{\text{turb}} , \quad \text{W/(kg}\cdot\text{K)} , \quad (\text{B.15})$$

where

$$\psi_{\text{gas}} = \frac{1}{\alpha \rho T} \nabla \cdot (\alpha \kappa \nabla T) \quad (\text{B.16})$$

and

$$\psi_{\text{particle}} = \begin{cases} -\frac{1}{\alpha \rho T} \frac{\dot{m}}{VG} s_p [h_c(T-\bar{T}) + h_r(T-\bar{T})] & , \text{ before ignition } , \\ 0 & , \text{ after ignition } , \end{cases} \quad (\text{B.17})$$

with

$$h_c = \frac{\kappa}{\frac{\dot{D}_p}{2}} + 0.2 \left(\frac{\gamma}{\gamma-1} \frac{R}{M} \frac{(\kappa 2 p)^2 |u-\bar{u}|}{u \dot{D}_p / 2} \right)^{1/3} , \quad \text{W/(m}^2 \cdot \text{K)} , \quad (\text{B.18})$$

and

$$h_r = \epsilon^* \sigma_{SB} (T+\bar{T}) (T^2+\bar{T}^2) , \quad \text{W/(m}^2 \cdot \text{K)} . \quad (\text{B.19})$$

In Eq. (B.19), ϵ^* is the particle emissivity, $\sigma_{SB} = 5.67032 \cdot 10^{-8} \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-4}$ is the Stephan-Boltzmann constant, and \bar{T} is the average grain surface temperature. The turbulent heat flux within the gas ψ_{gas} is given by Eqs. (B.36) and (B.37).

The acceleration term due to the drag between gas and particles is modeled by (Section 4.7.5)

$$A_{\text{drag}} = \begin{cases} A_{\text{Ergun}} , & \text{for } \alpha < 0.65 , \\ 4[(\alpha-0.65)A_{\text{Reynolds}} + (0.9-\alpha)A_{\text{Ergun}}] , & \text{for } 0.65 < \alpha < 0.9 , \\ A_{\text{Reynolds}} , & \text{for } 0.9 < \alpha , \end{cases} \quad (\text{B.20})$$

where

$$A_{\text{Ergun}} = (u-\bar{u}) \frac{\dot{a}_p}{v} \frac{2}{3} \frac{1}{\alpha} [1.75 |u-\bar{u}| + 150 (1-\alpha) \frac{\mu}{\rho \dot{D}_p}] , \quad \text{m/s}^2 \quad (\text{B.21})$$

and

$$A_{\text{Reynolds}} = (u-\bar{u}) \frac{\dot{a}_p}{v} [0.2 |u-\bar{u}| + 12 \frac{\mu}{\rho \dot{D}_p}] , \quad \text{m/s}^2 . \quad (\text{B.22})$$

The acceleration term due to intergranular stress is modeled by (Section 4.7.6)

$$A_{\text{stress}} = -a^{*2} \frac{1}{1-\alpha} \nabla(1-\alpha) , \quad \text{m}^2/\text{s}^2 , \quad (\text{B.23})$$

where $a^*(\alpha)$ is a sound speed function for the particulate phase. The function is modeled by

$$a^*(\alpha) = \begin{cases} a_{sp} \left(\frac{\alpha_1 - \alpha_0}{\alpha - \alpha_0} \right) \left(\frac{\alpha_2 - \alpha}{\alpha_2 - \alpha_1} \right) , & \text{for } \alpha_0 < \alpha < \alpha_2 , \\ 0 , & \text{for } \alpha_2 < \alpha . \end{cases} \quad (\text{B.24})$$

The burning rate is modeled by (Section 4.7.7)

$$\langle \dot{d} \rangle = B_0 + B_1 p^{B_2} , \quad \text{m/s} . \quad (\text{B.25})$$

The source term Γ is (Section 4.7.8)

$$\Gamma = \frac{1}{\alpha} \frac{\rho}{\rho} \frac{m}{VG} s_p \langle \dot{d} \rangle , \quad 1/\text{s} . \quad (\text{B.26})$$

The enthalpy factor H of the source term (Section 4.7.8) is defined by

$$H = \frac{1}{T} [(\hat{e} + p/\rho^*) - (e + p/\rho)] , \quad \text{J}/(\text{kg} \cdot \text{K}) , \quad (\text{B.27})$$

where \hat{e} is

$$\hat{e} = \frac{1}{\gamma-1} \frac{R}{M} T_{\text{flame}} = \frac{1}{\gamma-1} g_a I_p , \quad \text{J/kg} , \quad (\text{B.28})$$

with $g_a = 9.80665 \text{ m/s}^2$ being the standard acceleration.

The particle geometry enters the equations as the four functions $v_p(\dot{d})$, $s_p(\dot{d})$, $\dot{b}_p(\dot{d})$ and $a_p(\dot{d})$. We provide the formulas that define these functions for spherical, cylindrical, and tubular grains.

For a spherical grain with initial diameter \hat{D}_0 one defines

$$R = \max(0, (\hat{D}_0 - 2\hat{d})/2)$$

$$v_p = \frac{4}{3} \pi R^3 ,$$

$$s_p = 4\pi R^2 ,$$

$$a_p = \pi R^2 ,$$

$$\hat{D}_p = 2R .$$

} (B.29)

A solid cylindrical grain may be described by its initial diameter, \hat{D}_0 , and height, \hat{L}_0 . Let

$$R = (\hat{D}_0 - 2\hat{d})/2 ,$$

$$L = \hat{L}_0 - 2\hat{d} .$$

} (B.30)

If either $R < 0$ or $L < 0$, then the grain has been burnt. If both quantities are positive, then we define

$$v_p = \pi R^2 L ,$$

$$s_p = 2\pi R(R+L) ,$$

$$a_p = (2RL + \pi R^2)/2 ,$$

$$\hat{D}_p = (2R+L)/2 .$$

} (B.31)

A tubular grain may be defined by its initial height L_0^* and the initial outer and inner diameters, D_0^* and d_0^* , respectively. Let

$$\left. \begin{aligned} R &= (D_0^* - d_0^*)/2 , \\ r &= (d_0^* + 2d^*)/2 , \\ L &= L_0^* - 2d^* . \end{aligned} \right\} \quad (B.32)$$

The grain is completely burnt if either $R-r < 0$ or $L < 0$. If both of these quantities are positive, then the grain geometry functions are

$$\left. \begin{aligned} v_p &= \pi(D_0^* + d_0^*) (R-r)L/2 , \\ s_p &= \pi(D_0^* + d_0^*) (R-r+L) , \\ a_p &= (2RL + \pi(R^2 - r^2))/2 , \\ \bar{D}_p^* &= (2R+L)/2 . \end{aligned} \right\} \quad (B.33)$$

We consider a detailed study of turbulence models for interior ballistics flows to be outside the scope of this report. Hence, the correlation models are quite elementary and are listed in this report only for completeness. The acceleration by the gas phase turbulent stress tensor A_{turb} and the turbulent heat dissipation function ϕ_T , could have the same form as A_{visc} (Eq. (B.9)) and $\bar{\phi}(E)$, (Eq. (B.12)), respectively, but with different viscosity coefficients, that is,

$$A_{turb} = \frac{1}{\alpha\rho} \nabla \cdot \{ \alpha [2 \mu_T E + (\lambda_T - \frac{2}{3} \mu_T) (\text{trace } E) I] \} , \quad \text{m/s}^2 \quad (B.34)$$

$$\phi_T = 2 \mu_T \text{trace}(E^2) + (\lambda_T - \frac{2}{3} \mu_T) (\text{trace } E)^2 , \quad \text{W/m}^3 \quad (B.35)$$

and μ_T and λ_T denote the viscosity coefficients for turbulent flows. The manner in which these coefficients are determined strongly depends on the particular turbulence model one uses and, hence, will not be given. As

discussed in Section 4.7.6, the solid phase turbulent stress tensor $\hat{\pi}_T^*$ is set to zero. The turbulence heat flux vector Q_T is modeled by Ishii (1975) and Gibeling et al. (1980) as

$$Q_T = -\kappa_T \left[\nabla T - \frac{\nabla \alpha}{\alpha} (T_1 - T) \right] , \quad W/m^2 , \quad (B.36)$$

where T_1 is an average temperature on the interface (a function of T and \hat{T}) and κ_T is given by an algebraic formula involving an effective viscosity and Prandtl number. Consequently, ∇_{turb} in Eq. (B.15) is modeled as

$$\nabla_{turb} = -\frac{1}{\alpha \rho T} \nabla \cdot (\alpha Q_T) , \quad W/(kg \cdot K) . \quad (B.37)$$

FREE BOUNDARY PROBLEMS WITH NONLINEAR SOURCE TERMS

Gunter H. Meyer
School of Mathematics
Georgia Institute of Technology
Atlanta, Georgia 30332

Abstract. Multi dimensional free boundary problems with nonlinear reaction terms are approximated with a modified method of lines. An iterative numerical method results in which only one dimensional free boundary problems are solved sequentially. The algorithm is applied to an obstacle problem, a Michaelis-Menten reaction problem and a two component second order reaction problem.

The Algorithm. The term "free boundary problem" is used to describe a boundary value problem for differential equations where the domain of definition of the dependent variable is unknown a priori and must be determined simultaneously with the solution of the equations. Solidification, ablation, free streamline and some shock problems are common examples of free boundary and interface problems. Front tracking methods for free boundary problems describe those solution methods which specifically use the geometry of the free boundary in the solution algorithm. Several survey papers on free boundary problems have been published which detail their origin and the various solution methods (see, e.g. [2]).

In this report we shall examine front tracking for elliptic and parabolic free boundary problems involving nonlinear reactions. We shall employ the method of lines, which leads to a sequence of one dimensional free boundary problems from which the solution of the multi dimensional problem is determined. A description of the method of lines for linear differential equations appears in [4], and a mathematical analysis of the numerical algorithm is given in [5] for the Reynolds equation of hydrodynamic lubrication.

It is possible to extend the results of [5] by means of monotonicity arguments to certain nonlinear differential equations of the form

$$\begin{aligned} (1.1) \quad & \Delta u = f(u, \vec{x}) \\ \text{subject to} \quad & \\ (1.2) \quad & u = \partial u / \partial n = 0 \end{aligned}$$

on the free boundary. In particular, it is required that

$$f(u, \vec{x}) = f_1(u, \vec{x}) + f_2(u, \vec{x})$$

where f_1 is uniformly bounded and

$$\frac{\partial f_2}{\partial u} > -\lambda_0$$

where λ_0 is the first eigenvalue of the smallest domain into which the computed domain can be imbedded. Details of the proof will be given elsewhere.

This research was supported by the U.S. Army Research Office under Contract DAAG-79-0145

It is the purpose of this note to identify some problems which fit into this setting, and to examine the performance of the proposed numerical methods for such problems.

Specifically, we shall consider the following problem on the unit square R

$$(1.3) \quad \begin{aligned} \Delta u &= f(u, x, y) & (x, y) \in D \\ u &= g(x, y) & (x, y) \in \partial D \\ u &= 0 \\ \partial u / \partial n &= 0 \end{aligned} \quad \left. \vphantom{\begin{aligned} \Delta u &= f(u, x, y) \\ u &= g(x, y) \\ u &= 0 \\ \partial u / \partial n &= 0 \end{aligned}} \right\} \quad y = s(x)$$

where D is the subset of the unit square which bounded above by the free boundary $y=s(x)$. On the bottom and on the sides of the square below the free boundary either Dirichlet or Neumann conditions are prescribed. We remark that the restriction to the Laplacian is not essential. The convergence proof applies to more general elliptic operators without cross derivatives, while the numerical results indicate that the method works equally well in the presence of cross derivatives. Hence a combination of the method of lines and domain mapping methods may be considered to free the method of lines from some of the geometric restrictions which so far have had to be observed.

The free boundary problem (1.3) is the classical obstacle problem. It arises, for example, when a membrane of altitude w and supported on the boundary of the square is pushed up by an obstacle v . In the usual obstacle problem the shape of the obstacle is prescribed. In (1.3) the shape would depend on the membrane itself. Where the membrane does not touch the obstacle it satisfies Laplace's equation. At the point of contact it assumes the same altitude and slope as the obstacle. The function $u=w-v$ then satisfies a problem like (1.3). As is well known, the obstacle problem is usually written and analyzed as a variational inequality (see, e.g. [3]). Our numerical method makes no assumption on the structure of (1.3), but our convergence proof does.

The numerical solution of (1.3) can proceed as follows. On the basis of the maximum principle (or physical reasoning) one can often find an upper and lower bound on the solution of (1.3), say $|u| \leq C$. Then let K be a constant chosen so large that

$$(1.4) \quad \left| \frac{\partial f(u, x, y)}{\partial u} \right| \leq K \quad |u| \leq C, \quad (x, y) \in R.$$

Now the following algorithm is suggested in [2, p.370] for the nonlinear Poisson equation

$$(1.5) \quad \Delta u^k - Ku^k = f(u^{k-1}, x, y) - Ku^{k-1}.$$

This iteration can be combined in a natural way with the line-SOR approach used in [4] for free boundary problems. Specifically, let

$0=x_0 < x_1 < \dots < x_{n+1} = 1$ denote an equidistant partition and let u_{xx} be replaced by finite differences such as

$$(1.6) \quad u_{xx}(x_i, y) \approx \frac{u_{i+1} + u_{i-1} - 2u_i}{\Delta x^2}$$

or

$$(1.7) \quad u_{xx}(x_i, y) \approx \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}}{12\Delta x^2}$$

Then for $i=1, \dots, N$ and $k=1, 2, \dots$ we solve repeatedly the one dimensional problem

$$(1.8) \quad \begin{aligned} L\bar{u} &= \bar{u}'' - (K+a_i)\bar{u} = F_1^k(y) = 0 \\ \bar{u}(0) &= g(x_1, 0), \quad \bar{u}(s_1^k) = \bar{u}'(s_1^k) = 0 \\ u_i^k &= u_i^{k-1} + \omega(u - u_i^{k-1}) \end{aligned}$$

where

$$\begin{aligned} a_i &= \frac{2}{\Delta x^2} \\ F_1^k(y) &= -\frac{u_{i+1}^{k-1} + u_{i-1}^k}{\Delta x^2} + f(u_i^{k-1}, y) - Ku_i^{k-1} \end{aligned}$$

if (1.6) is used, or

$$\begin{aligned} a_i &= \frac{30}{12\Delta x^2} \\ F_1^k(y) &= \frac{u_{i+2}^{k-1} - 16u_{i+1}^{k-1} - 16u_{i-1}^k + u_{i-2}^k}{12\Delta x^2} + f(u_i^{k-1}, y) - Ku_i^{k-1} \end{aligned}$$

if (1.7) is used. On the boundaries x_0 and x_{N+1} the prescribed Dirichlet data or reflection (Neumann) data are used.

The one dimensional problem (1.8) can be solved in a variety of ways. Here we use the invariant imbedding method outlined in [4]. Briefly, we write

$$\bar{u} = R_1^k(y)\bar{u}' + w_1^k(y)$$

where R and w are the (assumed known or computable) solutions of the well defined initial value problems

$$(1.9) \quad \begin{aligned} R_1' &= 1 - (K+a_i)R_1^2, & R_1(0) &= 0 \\ \text{and} & & & \\ w_1^k &= -(K+a_i)R_1 w_1^k - R_1(y)F_1^k(u), & w_1^k(0) &= g(x_1, 0). \end{aligned}$$

The free boundary s_1^k is chosen as the smallest solution of

$$(1.10) \quad \phi_1^k(y) = w_1^k(y) = 0.$$

If no solution exists on $(0, 1]$ then we set $s_1^k = 1$. On the now given interval $(0, s_1^k)$ the function u_1 is then determined as the solution of the standard two point boundary value problem

$$Lu = F_1^k(y)$$

subject to given conditions at the end points.

The convergence proof of [5] applies to this problem under specific assumptions on f and g when $\omega = 1$. However, we generally choose $\omega > 1$ in order to accelerate convergence. We also note that in terms of programming and computing complexity and effort the above algorithm for nonlinear Poisson's equations differs only slightly from the corresponding algorithm for linear equations as described in [4].

Most of the numerical experiments carried out so far have used the second order approximation (1.6) and the trapezoidal rule for the integration of the equations (1.9) (as well as for the reverse sweep used for the computation of \bar{u}' and hence \bar{u} - for details see [4]). However, numerical experiments with the unstable Hele-Shaw flow problem [6] strongly indicate that the fourth order quotient should be used where possible. The primary reason for this suggestion is that the number of lines which can be handled economically with regard to storage and rate of convergence tends to be small compared to the number of mesh points per line. Therefore the accuracy of the approximation for u_{xx} tends to be too low when the second order method is used. We also have not yet experimented with a higher order integrator for the initial value problem (1.9), but work along these lines is planned in view of the relative sensitivity of the computed results to the number of mesh points along each line.

Numerical Examples. 1) A linear obstacle problem: Suppose an elastic membrane w covering the unit square is displaced by an ellipsoidal punch with the shape

$$v(x,y) = 1 - x^2 - 4(y-1)^2.$$

Then the difference $u = w - v$ between the membrane and the punch satisfies for $u > v$ the free boundary problem

$$(2.1) \quad \begin{aligned} \Delta u &= 10 & (x,y) \in D \\ u &= -v & (x,y) \in \partial D \end{aligned}$$

and

$$\bar{u} = \partial u / \partial n = 0 \quad y = s(x).$$

This is the classical obstacle problem for which the convergence of the numerical method and the convergence of the discrete solution to the continuous solution can be established a priori. A numerical solution of this problem with the second order approximation to u_{xx} is shown in Fig. 1. We note that in this case $K=0$. A nonlinear obstacle problem results if, for example, the membrane is simultaneously subject to a nonlinear force $f(w,x,y)$.

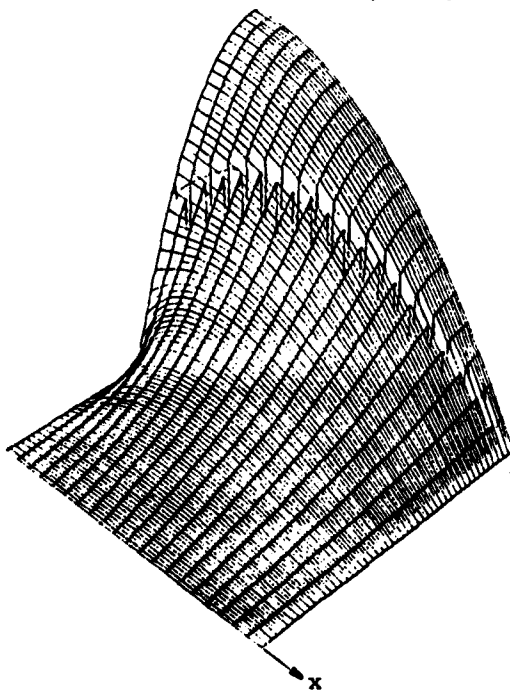


Fig. 1. Plot of membrane and obstacle for problem (2.1). $\Delta x = 1/20$, $\Delta y = 1/100$, $\omega = 1.6$; 51 iterations for a convergence criterion of $\max |u_i^k - u_i^{k-1}| \leq 10^{-8}$ between successive iterations. Total computing time 60 sec on the Cyber 170/700.

2) A Michaelis-Menten reaction problem: As a first example of a free boundary problem with a nonlinear source term we shall consider an extension of the oxygen diffusion-consumption model which is representative for a number of biological diffusion processes (see, e.g. [7]). Here an agent at concentration $u(x,y,t)$ is diffusing into the medium D (such as oxygen diffusing through living tissue). As it diffuses it is consumed at a non-constant rate according to a Michaelis-Menten reaction. The concentration may then be described by the free boundary problem

$$(2.2) \quad \Delta u - cu_t = f(u, x, y)$$

with

$$f(u, x, y) = \frac{\alpha u}{1 + u} + \epsilon(x, y)$$

where $\epsilon(x, y)$ is a local threshold consumption rate. At the free boundary the concentration and its gradient vanish. (Other conditions could be imposed, such as threshold concentrations or gradients - see the reaction problem below.) If problem (2.2) is time discretized and solved as a sequence of time implicit elliptic equations then the monotone convergence theory applies at every time step.

For numerical work we shall use the same geometry as in the obstacle problem. Specifically we shall assume that

$$\epsilon(x, y) = 8(x-0.5)^2 \quad (x, y) \in D$$

$$\partial u / \partial n = 0 \text{ on } x = 0 \text{ and } x = 1$$

$$u(x, 0) = x(1-x).$$

If $s(x)=1$ then we assume that $\partial u / \partial y = 0$. For convenience we shall consider only the steady state case ($c=0$). Fig. 2 shows D and the free boundary. From the data symmetry about $x=0.5$ is expected although it is not specifically used in the program.

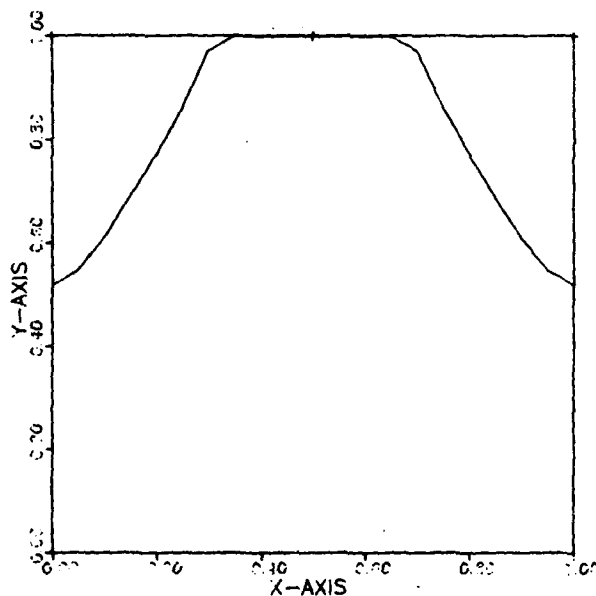


Fig. 2. Plot of the steady state free boundary for the Michaelis-Menten reaction problem (2.2). $\Delta x=1/20$, $\Delta y=1/100$, $\omega=1.6$, 49 iterations. The same number of iterations were required for $K = \alpha = 1$ as for $K = \alpha = 0$.

3) Second order diffusion-reaction: As a third example let us consider a two-component reaction problem where a substance at concentration u diffuses into an immobile substance at concentration v while undergoing a second order irreversible reaction with it. The model equations and an application to the diffusion of oxygen in nickel are discussed in [8] where an asymptotic formula for the diffusion front is developed in one space dimension (which, however, does not correspond to a free boundary). Here we shall deal with a two dimensional problem in which movement into unreacted zones can occur only if the gradient $\partial u / \partial n$ on the diffusion front exceeds a given threshold value.

Specifically, for the same geometry as in examples 1) and 2) we shall consider the time dependent boundary value problem

$$(2.3) \quad \begin{aligned} u_t &= \Delta u - kuv \\ v_t &= -kuv \\ \partial u / \partial n &= \partial v / \partial n = 0 \quad \text{on } x = 0 \text{ and } x = 1, \quad t > 0 \\ u(x, y, 0) &= 0 \\ v(x, y, 0) &= v_0(x, y) \end{aligned} \quad x \in (0, 1), \quad 0 < y < s(x, t)$$

and the free boundary condition

$$u = 0 \text{ and } |\partial u / \partial n| \geq \epsilon > 0 \quad y = s(x, t).$$

This problem is readily converted into a single variable problem for u because

$$v(x, y, t) = v_0(x, y) \exp(-k \int_0^t u(x, y, r) dr).$$

Thus, we shall consider the scalar equation

$$\Delta u - u_t = kuv_0 \exp(-k \int_0^t u dr)$$

subject to the appropriate boundary conditions induced by (2.3).

A fully time implicit approximation based on a backward difference quotient for u_t and the trapezoidal rule for the integral then leads to the sequence of elliptic problems at time t_n for $u \equiv u_n$

$$(2.4) \quad \begin{aligned} \Delta u &= f(u, x, y, t) \\ \text{where} \quad f(u, x, y, t) &= \frac{u - u_{n-1}}{\Delta t} + kuv_0(x, y) \phi(x, y, t_{n-1}) \exp(-k \Delta t \frac{u + u_{n-1}}{2}) \end{aligned}$$

with

$$\phi(x, y, t_n) = \phi(x, y, t_{n-1}) \exp(-k \Delta t \frac{u_n + u_{n-1}}{2})$$

For an input concentration of

$$u(x, y, 0) = (t / (1+t)) (0.1 + 16x^2(1-x)^2)$$

it follows immediately on physical grounds (or from the maximum principle) that

$$0 \leq u \leq 1.1t / (1+t), \quad 0 \leq v \leq v_0.$$

Since $\phi(x, y, t_n) \leq 1$ it is simple to check that

$$|\frac{\partial f}{\partial u}| \leq \frac{1}{\Delta t} + k$$

for sufficiently small Δt . Hence for the constant K in (1.4) we shall choose

$$K = \frac{1}{\Delta t} + k.$$

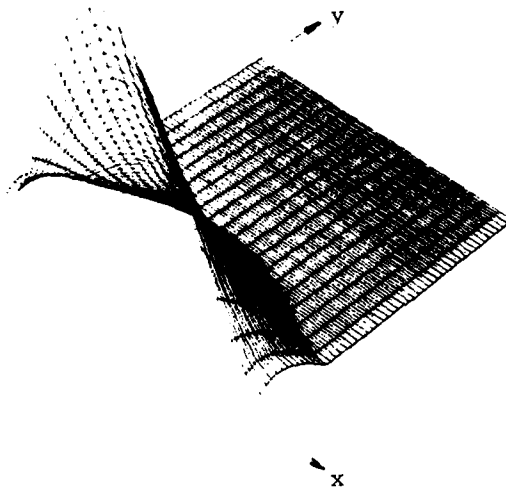


Fig. 3. Plot of $u(x, y, t)$. $\Delta x = 1/20$, $\Delta y = 1/100$, $t = 0.1$, $\Delta t = 0.1/20$, $\omega = 1.6$. About 34 iterations per time step are required for convergence. Total computing time for 20 time steps - 240 sec on the Cyber 170/700.

Some final comments. The most restrictive aspect of the method of lines as discussed in the literature is the strong dependence on rectangular or circular regions. But as is well known (see, e.g. [9]) irregular regions can often be mapped onto regular computational domains at the expense of complicating the differential equation. In order to apply the method of lines over the computational domain its behavior for general elliptic equations of the form

$$\sum_{i,j} a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_i b_i \frac{\partial u}{\partial x_i} + cu = f$$

must be established.

First experiments with the method of lines over a rectangular computational domain show little change in its performance when the equation no longer is in divergence form, although fourth order approximations for u_{xx} and $(u_y)_x$ appear to be important. The application of the method of lines to free boundary problems on irregular domains is currently under consideration.

REFERENCES

1. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience, N. Y., 1962.
2. A. Fasano and M. Primicerio, eds., *Free Boundary Problems: Theory and Applications*, Montecatini 1981, to appear as Springer Notes.
3. D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and their Application*, Academic Press, New York, 1980.
4. G. H. Meyer, The method of lines and invariant imbedding for elliptic and parabolic free boundary problems, *SIAM J. Num. Anal.* 18 (1981), 150-164.
5. _____, An analysis of the method of lines for the Reynolds equation in hydrodynamic lubrication, *SIAM J. Num. Anal.* 18 (1981), 165-177.
6. _____, Hele-Shaw flow with a cusping free boundary, *J. Comp. Phys.* 44 (1981), 262-276.
7. J. Ockendon and W. Hodgkins, eds., *Moving Boundary Problems in Heat Flow and Diffusion*, Clarendon Press, Oxford, 1975.
8. G. Roberts, Diffusion with chemical reaction, *Metal Science*, Feb. 1979, 94-97.
9. Workshop on Numerical Grid Generation, NASA Langley Notes, 1980.

NUMERICAL SOLUTION TO AN AUTOFRETTAGED TUBE
WITH CONSTRAINING WALLS AND END CLOSURES

Peter C. T. Chen
U.S. Army Armament Research and Development Command
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189

ABSTRACT. This paper presents a numerical study of a container autofrettage process. This process uses internal hydraulic pressure to expand the tube, restraining containers to control the amount of tube expansion and the press force to hold the end closures. The incremental finite-difference approach developed recently by the author is extended to obtain numerical results. The effect of restraining walls and the press force on the displacements and stresses are discussed.

1. INTRODUCTION. The importance of favorable residual stresses in an autofrettaged tube is well known (ref. 1). The container method is one of the autofrettage processes currently being used for gun tubes. It uses internal hydraulic pressure to expand the tube. Restraining containers or dies are used to control the amount of tube expansion by means of a small, predetermined clearance between the inside of the containers and the outside of the tube. The press is used to simply hold the end closures or seals in the ends of the tube and to support the forces of the internal pressure on the closures.

Many methods for solving the partially autofrettaged problem in a gun tube have been reported (refs. 2-6). However, the effect of constraining walls and end closures on the residual stresses have never been discussed. This paper presents a numerical study of the container autofrettage process. The finite difference approach developed recently by the author (ref. 6) is extended to obtain the numerical results. The material is assumed to obey the Mises' yield criterion and the Prandtl-Reuss incremental stress-strain relations.

2. FINITE-DIFFERENCE FORMULATION. Consider a long, open-end thick-walled cylinder of inner radius a and external radius b . The inside surface of the tube is subjected to hydraulic pressure p and an end force ($p\pi a^2$) is applied to simply hold the end closures or seals. The additional force f on the end closures will press against the tube. The amount of tube expansion is restricted by means of restraining containers of inside radius C . The cross section of the tube is divided into n rings with $r_1 = a, r_2, \dots, r_k = \rho, \dots, r_{n+1} = b$, where ρ is the radius of the elastic-plastic interface. Since the material behavior is nonlinear, an incremental approach is used. At the beginning of each incremental loading, the distribution of displacements, strains, and stresses are assumed to be known and we want to determine $\Delta u, \Delta \epsilon_r, \Delta \epsilon_\theta, \Delta \epsilon_z, \Delta \sigma_r, \Delta \sigma_\theta, \Delta \sigma_z$ at all grid points. According to the Prandtl-Reuss flow theory, the incremental stresses are related to the incremental strains by

$$\{\Delta\sigma_i\} = [d_{ij}] \{\Delta\epsilon_j\} \text{ for } i, j = r, \theta, z \quad (1)$$

and

$$[d_{ij}] = 2G[\nu/(1-2\nu) + \delta_{ij} - \sigma_i' \sigma_j' / s] \quad (2)$$

where

$$2G = E/(1+\nu) \quad , \quad S = \frac{2}{3} \left(1 + \frac{1}{3} H'/G\right) \sigma^2 \quad , \quad H'/E = \alpha/(1-\alpha) \quad ,$$

$$\sigma_m = (\sigma_r + \sigma_\theta + \sigma_z)/3 \quad , \quad \sigma_i' = \sigma_i - \sigma_m \quad ,$$

$$\sigma = (1/\sqrt{2})[(\sigma_r - \sigma_\theta)^2 + (\sigma_\theta - \sigma_z)^2 + (\sigma_z - \sigma_r)^2]^{1/2} > \sigma_0 \quad (3)$$

E is Young's modulus, ν is Poisson's ratio, δ_{ij} is the Kronecker delta, αE is the slope of the effective stress-strain curve, and σ_0 is the yield stress in simple tension or compression. When $\sigma < \sigma_0$ or $d\sigma < 0$, the state of stress is elastic and the last term in Eq. (2) disappears. Since the incremental stresses are related to the incremental strains by Eq. (1) and $\Delta u = r \Delta \epsilon_\theta$, there exists only three unknowns at each station that have to be determined for each increment of loading. Accounting for the fact that the axial strain ϵ_z is independent of r , the unknown variables in the present formulation are $(\Delta \epsilon_\theta)_i$, $(\Delta \epsilon_r)_i$, for $i = 1, 2, \dots, n, n+1$, and $\Delta \epsilon_z$.

The equation of equilibrium and the equation of compatibility are valid for both the elastic and the plastic regions of a thick-walled tube. The finite-difference forms of these two equations at $i = 1, \dots, n$ are given by (ref. 6)

$$\begin{aligned} & [(r_{i+1} - 2r_i)(d_{12})_i + (-r_{i+1} + r_i)(d_{22})_i](\Delta \epsilon_\theta)_i \\ & + [(r_{i+1} - 2r_i)(d_{11})_i + (-r_{i+1} + r_i)(d_{21})_i](\Delta \epsilon_r)_i \\ & + r_i(d_{12})_{i+1}(\Delta \epsilon_\theta)_{i+1} + r_i(d_{11})_{i+1}(\Delta \epsilon_r)_{i+1} \\ & + [(r_{i+1} - 2r_i)(d_{13}) + (-r_{i+1} + r_i)(d_{23})_i + r_i(d_{13})_{i+1}]\Delta \epsilon_z \\ & = (r_{i+1} - r_i)(\sigma_\theta - \sigma_r)_i - r_i[(\sigma_r)_{i+1} - (\sigma_r)_i] \end{aligned} \quad (4)$$

for the equation of equilibrium, and

$$\begin{aligned} & (r_{i+1} - 2r_i)(\Delta \epsilon_\theta)_i - (r_{i+1} - r_i)(\Delta \epsilon_r)_i + r_i(\Delta \epsilon_\theta)_{i+1} \\ & = (r_{i+1} - r_i)(\epsilon_r - \epsilon_\theta)_i - r_i[(\epsilon_\theta)_{i+1} - (\epsilon_\theta)_i] \end{aligned} \quad (5)$$

for the equation of compatibility.

3. BOUNDARY CONDITIONS AND INCREMENTAL LOADING. The three boundary conditions for the problem are

$$(1) \quad (d_{12})(\Delta\epsilon_\theta)_1 + (d_{11})_1(\Delta\epsilon_r)_1 + (d_{13})_1\Delta\epsilon_z = -\Delta p \quad (6)$$

$$(11) \quad (d_{12})_{n+1}(\Delta\epsilon_\theta)_{n+1} + (d_{11})_{n+1}(\Delta\epsilon_r)_{n+1} + (d_{13})_{n+1}\Delta\epsilon_z = 0 \quad (7a)$$

$$\text{before contact or } (\Delta\epsilon_\theta)_{n+1} = 0 \text{ after contact,} \quad (7b)$$

$$(111)$$

$$\sum_{i=1}^n (r_{i+1}-r_i) \{ r_i [(d_{23})_i(\Delta\epsilon_\theta)_i + (d_{13})_i(\Delta\epsilon_r)_i] + r_{i+1} [(d_{23})_{i+1}(\Delta\epsilon_\theta)_{i+1} + (d_{13})_{i+1}(\Delta\epsilon_r)_{i+1}] \} + \sum_{i=1}^n (r_{i+1}-r_i) [r_i(d_{33})_i + r_{i+1}(d_{33})_{i+1}] \Delta\epsilon_z = \Delta f / \pi \quad (8)$$

Now we can form a system of $2n+3$ equations for solving $2n+3$ unknowns, $(\Delta\epsilon_\theta)_i$, $(\Delta\epsilon_r)_i$, at $i = 1, 2, \dots, n, n+1$ and $\Delta\epsilon_z$. Equations (6), (7), and (8) are taken as the first and the last two equations, respectively, and the other $2n$ equations are set up at $i = 1, 2, \dots, n$ using Eqs. (4) and (5). The final system is an unsymmetric matrix of arrow type with the nonzero terms appearing in the last row and column and others clustered about the main diagonal, two below and one above.

In order to increase the efficiency of the program, an adaptive algorithm based on a scaled incremental-loading approach has been implemented. In each step, a dummy load-increment such as Δp is applied and the incremental results $\Delta\sigma_i$ for $i = r, \theta, z$ at all grids are determined. For all grid points at which $\sigma = ||\sigma_i|| < \sigma_0$, we compute the scalar g 's by the formula

$$g = \frac{1}{2} \{ \Gamma + [\Gamma^2 + 4 ||\Delta\sigma_i||^2 (\sigma_0^2 - ||\sigma_i||^2)^{1/2}] / ||\Delta\sigma_i||^2 \} \quad (9)$$

where

$$\Gamma = ||\sigma_i||^2 + ||\Delta\sigma_i||^2 - ||\sigma_i + \Delta\sigma_i||^2 \quad (10)$$

and $||\sigma_i||$, $||\Delta\sigma_i||$, $||\sigma_i + \Delta\sigma_i||$ are computed by

$$||\sigma_i||^2 = \frac{1}{2} [(\sigma_r - \sigma_\theta)^2 + (\sigma_\theta - \sigma_z)^2 + (\sigma_z - \sigma_r)^2] \quad (11)$$

Let λ be the minimum of the g 's. Then λ is the load-increment factor just sufficient to yield one additional point. A sequence of $\lambda^{(j)}$ can be determined for all steps $j = 1, 2, \dots, m$ and the updated results are

$$\begin{aligned} p(j) &= p(j-1) + \lambda(j) \Delta p(j) \\ \sigma_i(j) &= \sigma_i(j-1) + \lambda(j) \Delta\sigma_i(j), \text{ etc.} \end{aligned} \quad (12)$$

4. NUMERICAL RESULTS. The numerical results are obtained on the basis of the following parameters: $a = 1.895"$, $b = 3.21"$, $c = 3.2275"$, $n = 50$, $E = 30 \times 10^6$ psi, $\nu = 0.3$, $\sigma_0 = 17 \times 10^4$ psi and $H' = 0$. The maximum internal pressure applied is 13×10^4 psi (max. p) and the maximum end force applied against the tube is $f = -0.6 p \pi (b^2 - a^2)$. Introducing the dimensionless quantity $\bar{f} = f / [\pi (b^2 - a^2) p]$, we have $-0.6 < \bar{f} < 0$. Since the end force required to simply hold the end closures or scales is $-p \pi a^2$, the total end force applied on the end closures is $F = p \pi a^2 [-1 + \bar{f} (b^2/a^2 - 1)]$. In order to discuss the effect of end force f , the numerical results have been obtained for two extreme cases, i.e. $\bar{f} = 0, -0.6$.

(a) $\bar{f} = 0$. In this case the total end force applied on the end closures is $F = -p \pi a^2$, just enough to support the forces of the internal pressure on the closures. Therefore, there is no force applied at the end of the tube. The maximum internal pressure ($p = 13 \times 10^4$ psi) is applied incrementally in three different stages. The displacements u_a, u_b at the inside, outside surface as functions of internal pressure p are shown in Figure 1. In stage one, the elastic solution due to a dummy internal pressure is applied and the scaled factor to cause initial yielding is determined. The closed form elastic solution together with the Mises' yield criterion may be used and the pressure factor corresponding to initial yielding is $p^*/\sigma_0 = 0.36875$. In the second stage, scaled incremental-loading approach is used until the maximum allowable outside displacement ($c-b$) is reached. At the instant when the contact between the tube and container first occurs, the pressure p/σ_0 is 0.57916 and 96 percent of the tube has been yielded. In the third stage, there is no outside displacement and internal pressure is increased in 20 equal steps until the maximum $p/\sigma_0 = 0.76471$ has been reached. The relation between pressure and inside displacement is almost linear in this stage as shown in Figure 1. The results of the displacements at the end of three stages are represented by the points 1, 2, and 3. The corresponding results of the stress distributions for σ_r, σ_θ , and σ_z are shown in Figures 2 through 4, respectively. It can be seen that the stress distributions at the end of three loading stages are quite different. The residual stresses after unloading completely from the end of three loading stages have also been obtained. The results for the residual hoop stresses for three stages and the residual axial stress for the last stage are shown in Figure 5. The differences in residual stresses between stage 2 and 3 are much smaller than those before unloading. That is to say that further increase in internal pressure is possible in the presence of restraining container but the increased pressure makes little differences in the residual stresses. The purpose of the outside container is to prevent large displacements to occur.

(b) $\bar{f} = -0.6$. In this case the total end force applied on the end closures is $F = -p \pi a^2 (0.4 + 0.6 b^2/a^2)$. This end force is larger than that required to support the forces of the internal pressure on the closures. Therefore, the end force applied at the end of the tube is $f = -0.6 p \pi (b^2 - a^2)$. For this case the maximum internal pressure is applied incrementally in four different stages. The displacements u_a, u_b at the inside, outside surface as functions of internal pressure p are shown in Figure 6. The points 1 to 4 represent the corresponding results at the end of each loading stage. At the end of the first stage, initial yielding solution has been obtained and the pressure required is $p = 0.34594 \sigma_0$. In the second stage, 50 scaled

incremental-loading steps are applied until the entire tube becomes yielded. At the end of the second stage, the pressure factor p/σ_0 is 0.50194 and the outside displacement is still smaller than the clearance, i.e., $u_b = 0.86126 b \sigma_0/E < 0.0175$ ". Since the material is assumed to be ideally plastic, the tube would collapse if there were no outside restraining containers. A very small increase in internal pressure, say $\Delta p/\sigma_0 = 0.0001$, will close the clearance between the tube and container. The instant when the contact first occurs is called the end of loading stage 3. After the contact we increase the internal pressure in 29 equal steps until the maximum pressure has been reached. The relation between pressure and internal displacement is approximately linear in this stage as shown in Figure 6. The stress distributions for σ_r , σ_θ at the end of four loading stages are shown in Figures 7 and 8 respectively, and that for σ_z shown in Figure 4. The change in stresses during the third loading stage is too small to be shown graphically in these figures but the differences in displacements are large as shown in Figure 6. The residual stresses due to complete unloading from the end of each loading stage have also been obtained and some of the results are shown in Figure 9. It can be seen that the differences in stresses during loading stages three and four are quite large but the corresponding residual stresses are very close. This also shows that the effect of outside containers and end forces on the residual stresses is small but their effects on the displacement and stresses during loading are large. In the presence of the press force on the tube end, the axial stress distributions change drastically as shown in Figure 4 as compared with the case of no end force. By comparing the results for the residual axial stresses as shown in Figures 5 and 9, we can see two different stress patterns, one is almost the reverse of the other. As a result of extra press force on the tube end, the final residual stresses can change signs.

REFERENCES

1. Davidson, T. E. and Kendall, D. P., "The Design of Pressure Vessels for Very High Pressure Operation," Watervliet Arsenal Report WVT-6917. Also in Mechanical Behavior of Materials Under Pressure (edited by Pugh, H. L. D.), Elsevier Co., 1970, Chapter 2.
2. Hodge, P. G. and White, G. N., "A Quantitative Comparison of Flow and Deformation Theories of Plasticity," J. Appl. Mech., Vol. 17, 19509, pp. 180-184.
3. Chu, S. C., "A More Rational Approach to the Problem of an Elastoplastic Thick-Walled Cylinder," J. of the Franklin Institute, Vol. 294, 1972, pp. 57-65.
4. Chen, P. C. T., "The Finite Element Analysis of Elastic-Plastic Thick-Walled Tubes," Proc. of Army Symposium on Solid Mechanics, 1972, The Role of Mechanics in Design-Ballistic Problems, pp. 243-253.
5. Elder, A. S., Tomkins, R. C., and Mann, T. L., "Generalized Plane Strain in an Elastic, Perfectly Plastic Cylinder, With Reference to the Hydraulic Autofrettage Process," Trans. 21st Conference of Army Mathematicians, 1975, pp. 623-659.
6. Chen, P. C. T., "Generalized Plane-Strain Problems in an Elastic-Plastic Thick-Walled Cylinder," Trans. 26th Conference of Army Mathematicians, 1980, pp. 265-275.

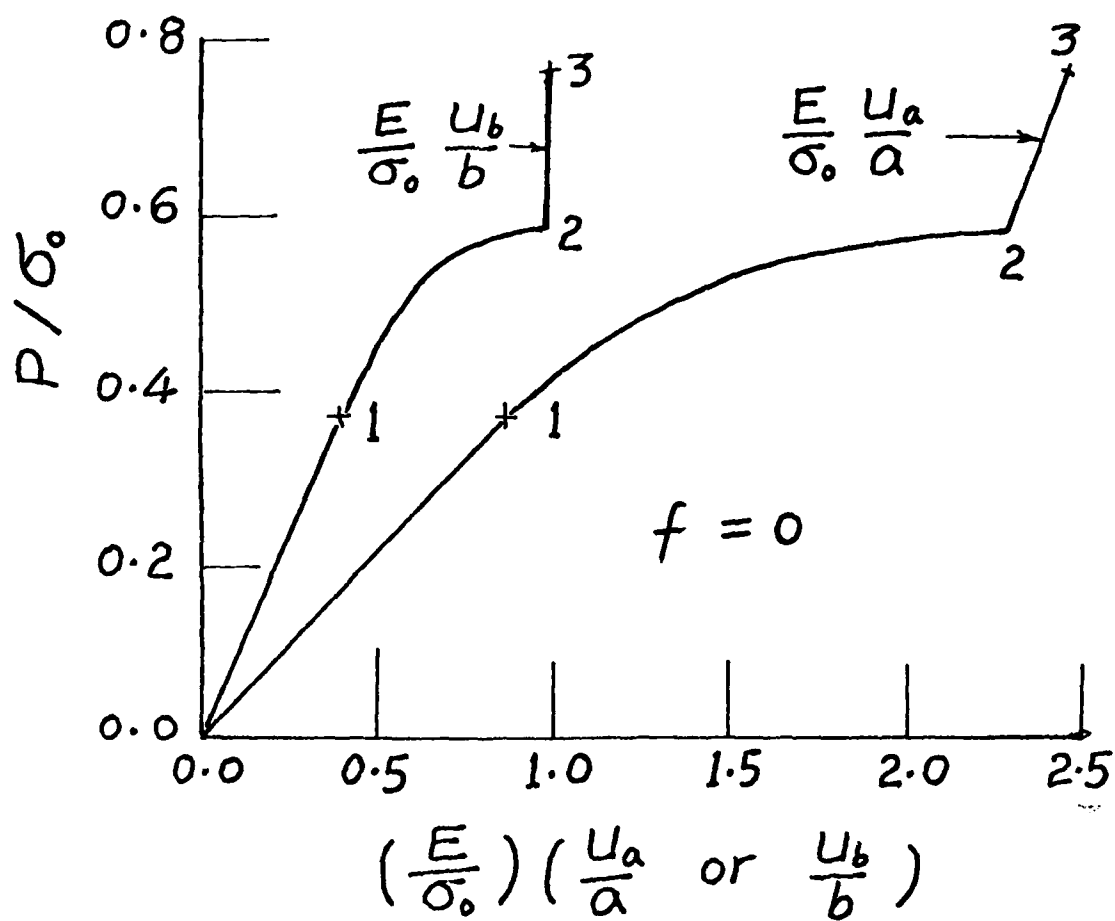


Figure 1. The boundary displacements u_a , u_b as functions of internal pressure p with no end force f .

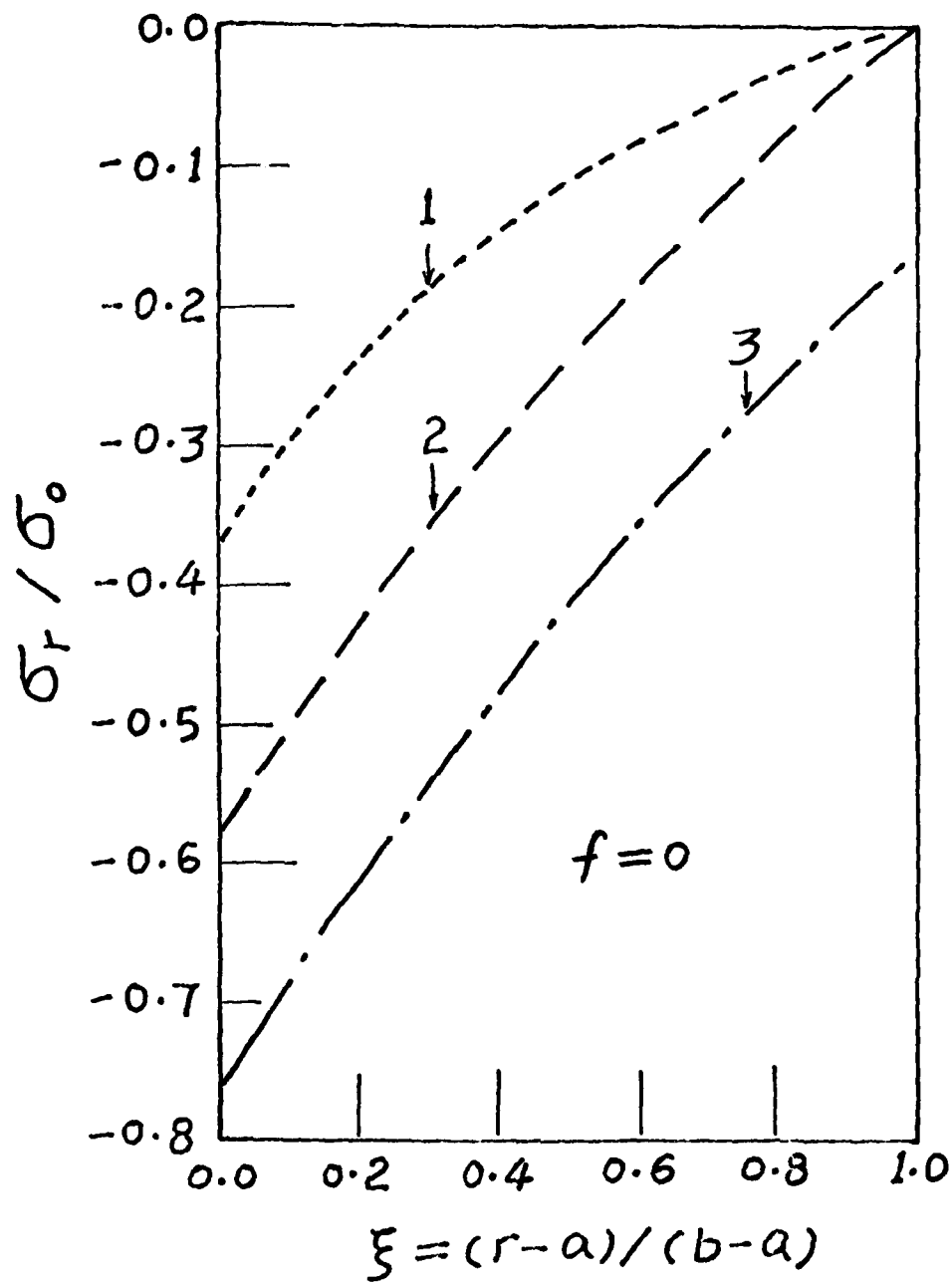


Figure 2. The radial stress distributions during loading with no end force f .

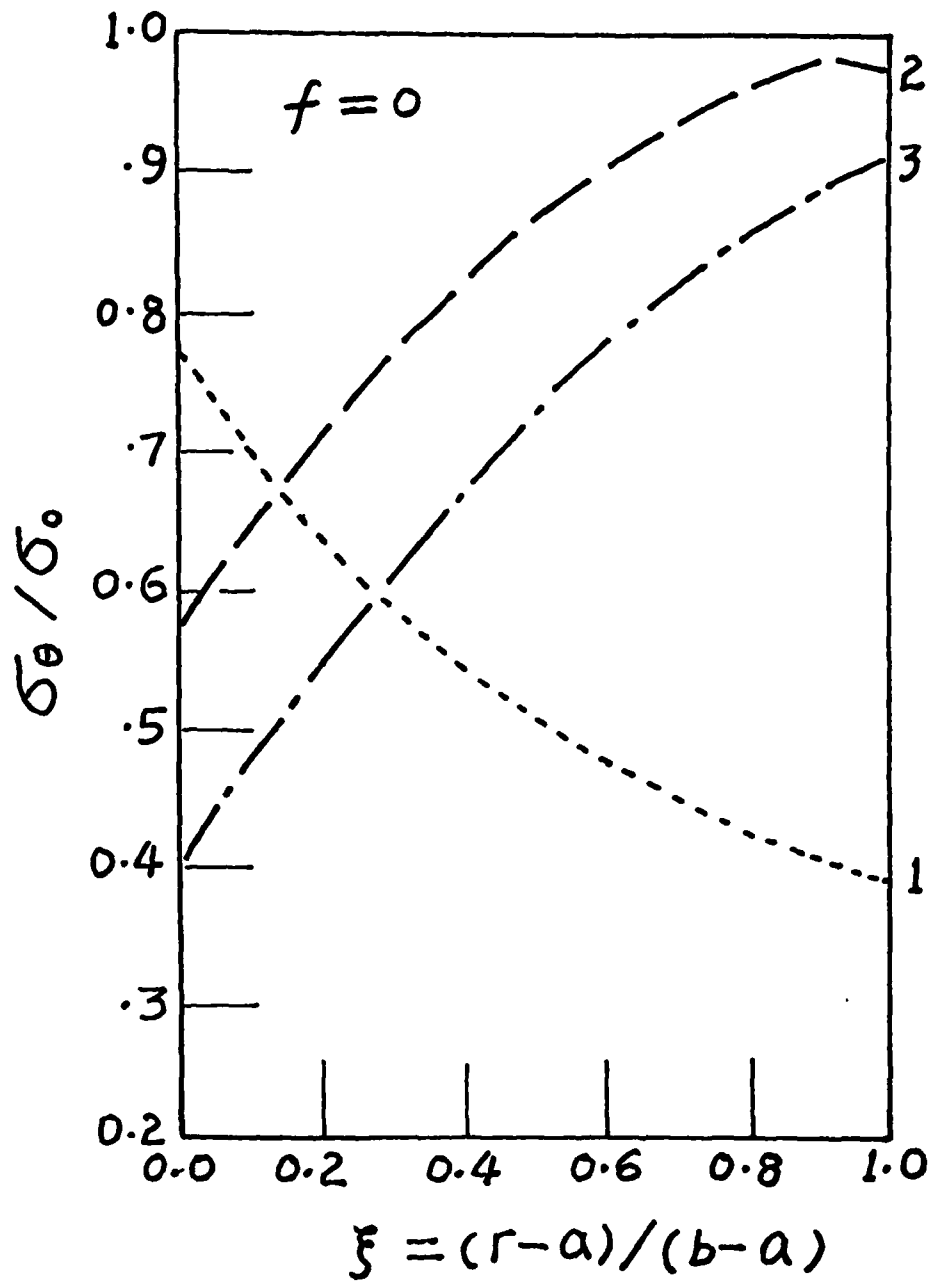


Figure 3. The hoop stress distributions during loading with no end force f .

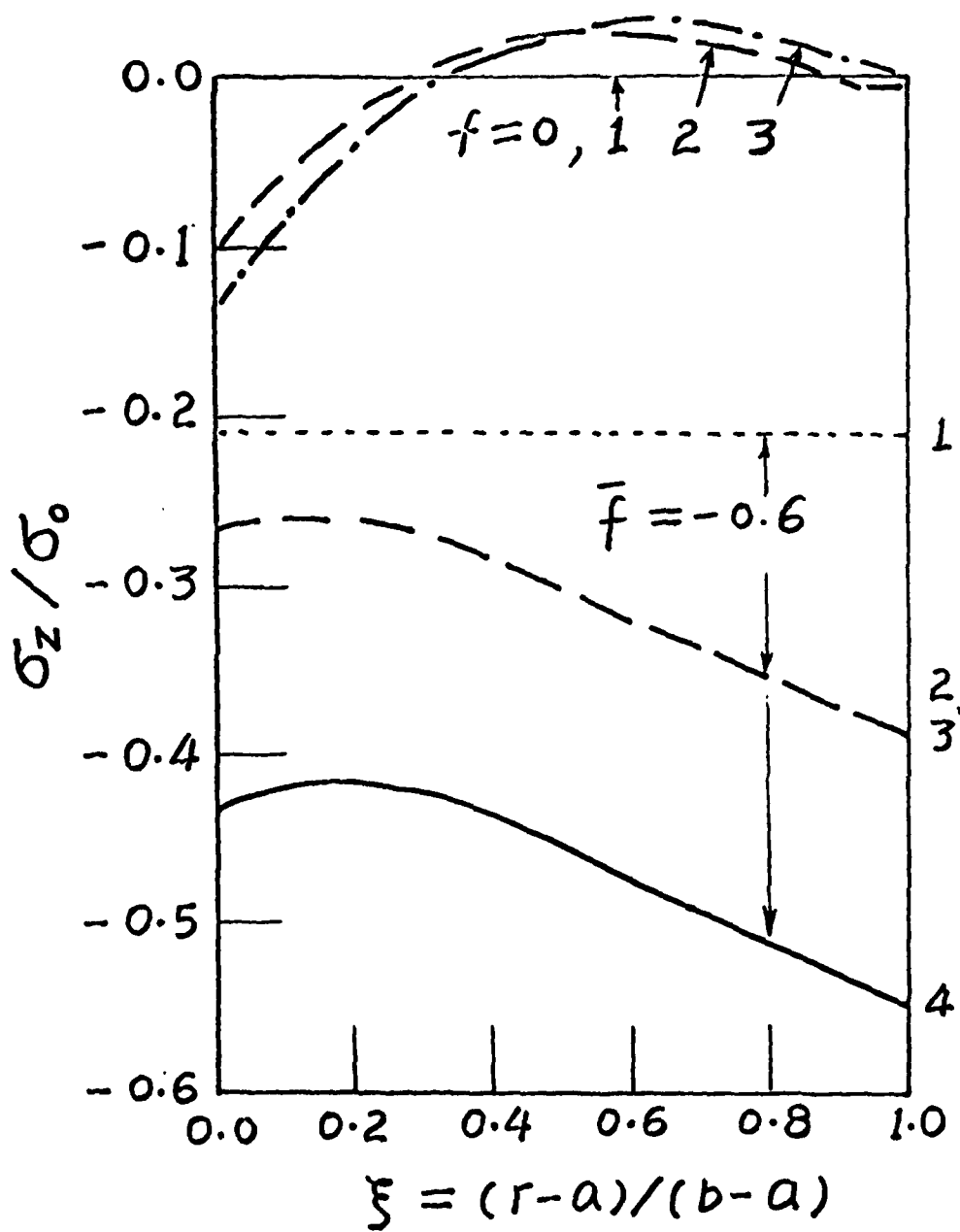


Figure 4. The axial stress distributions during loading with $\bar{f} = 0$ and -0.6 .

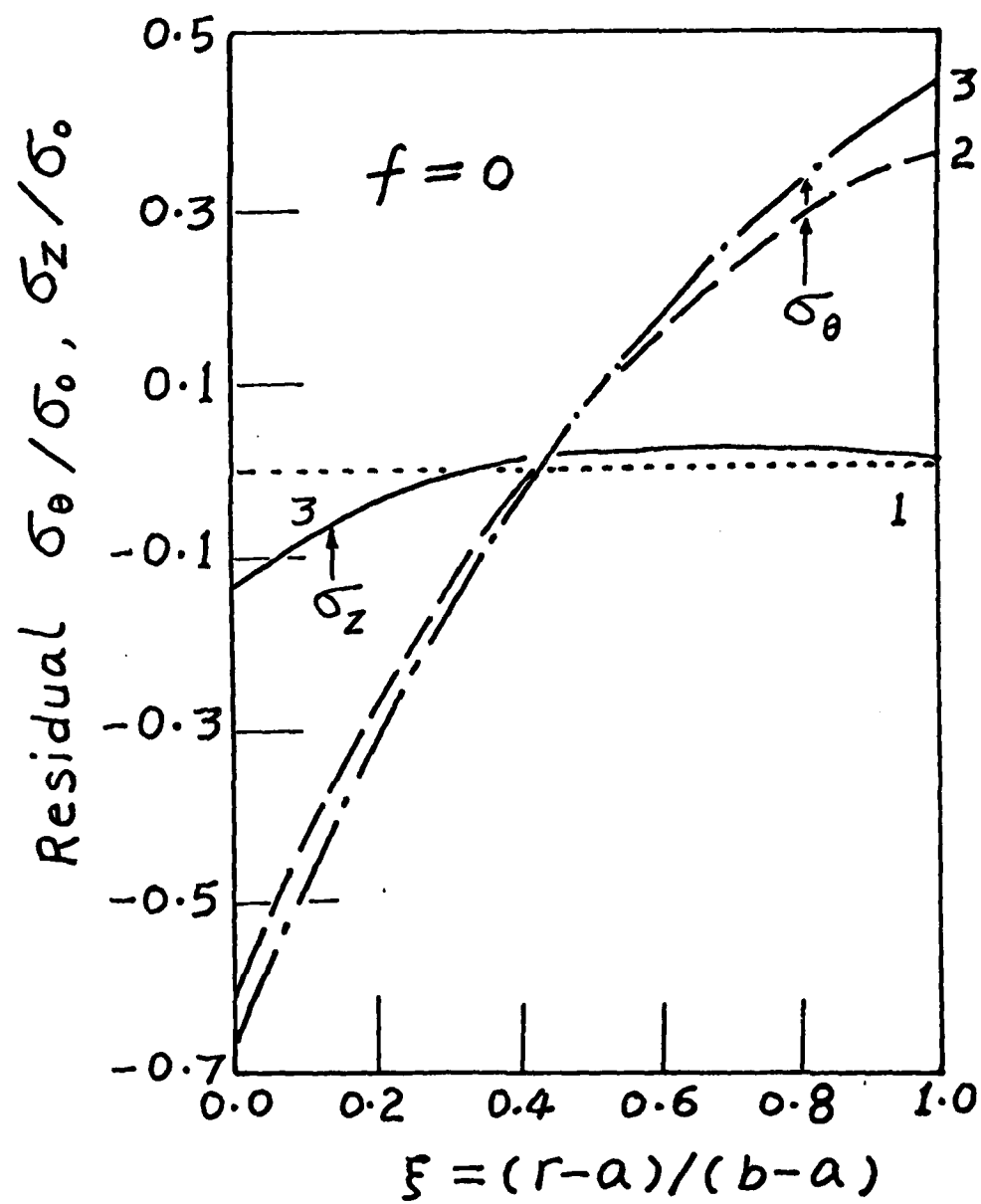


Figure 5. The residual stresses due to complete unloading from different stages with no end force.

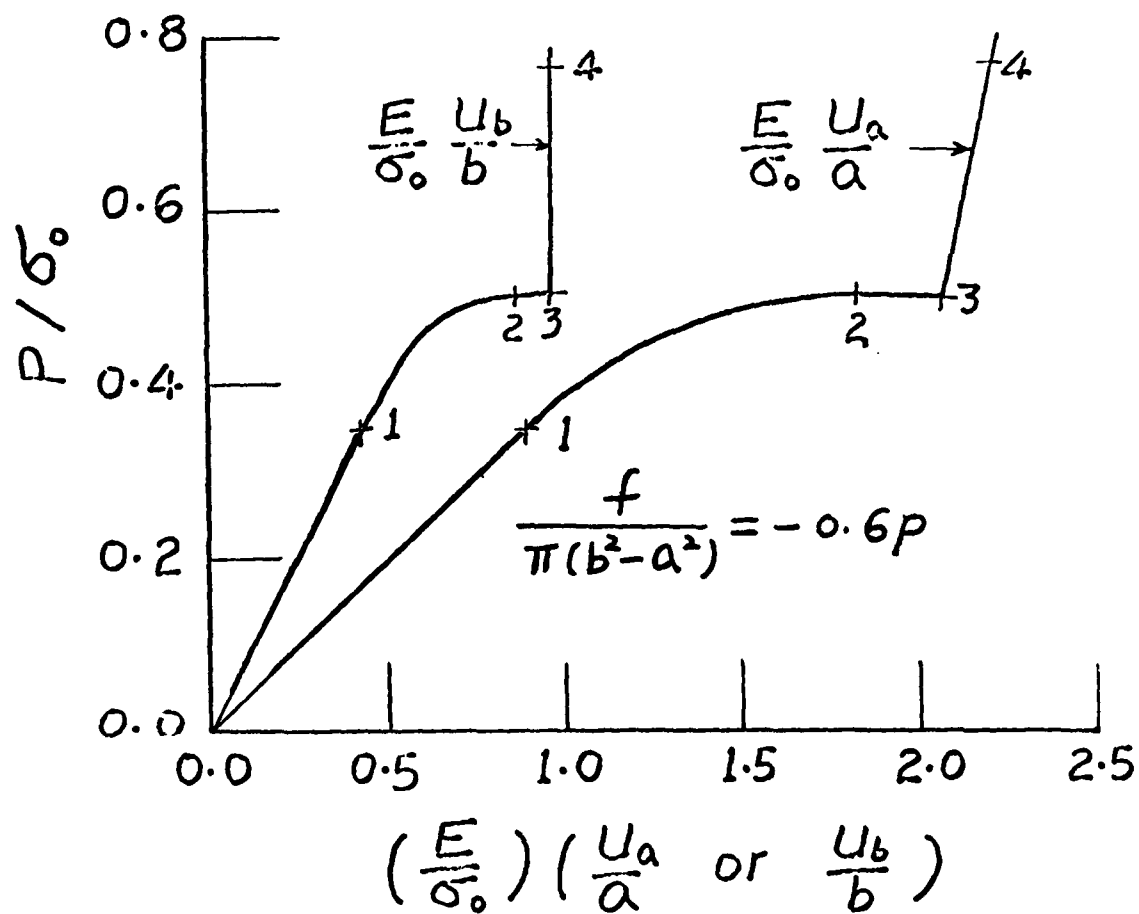


Figure 6. The boundary displacements u_a , u_b as functions of internal pressure p with $\bar{f} = -0.6$.

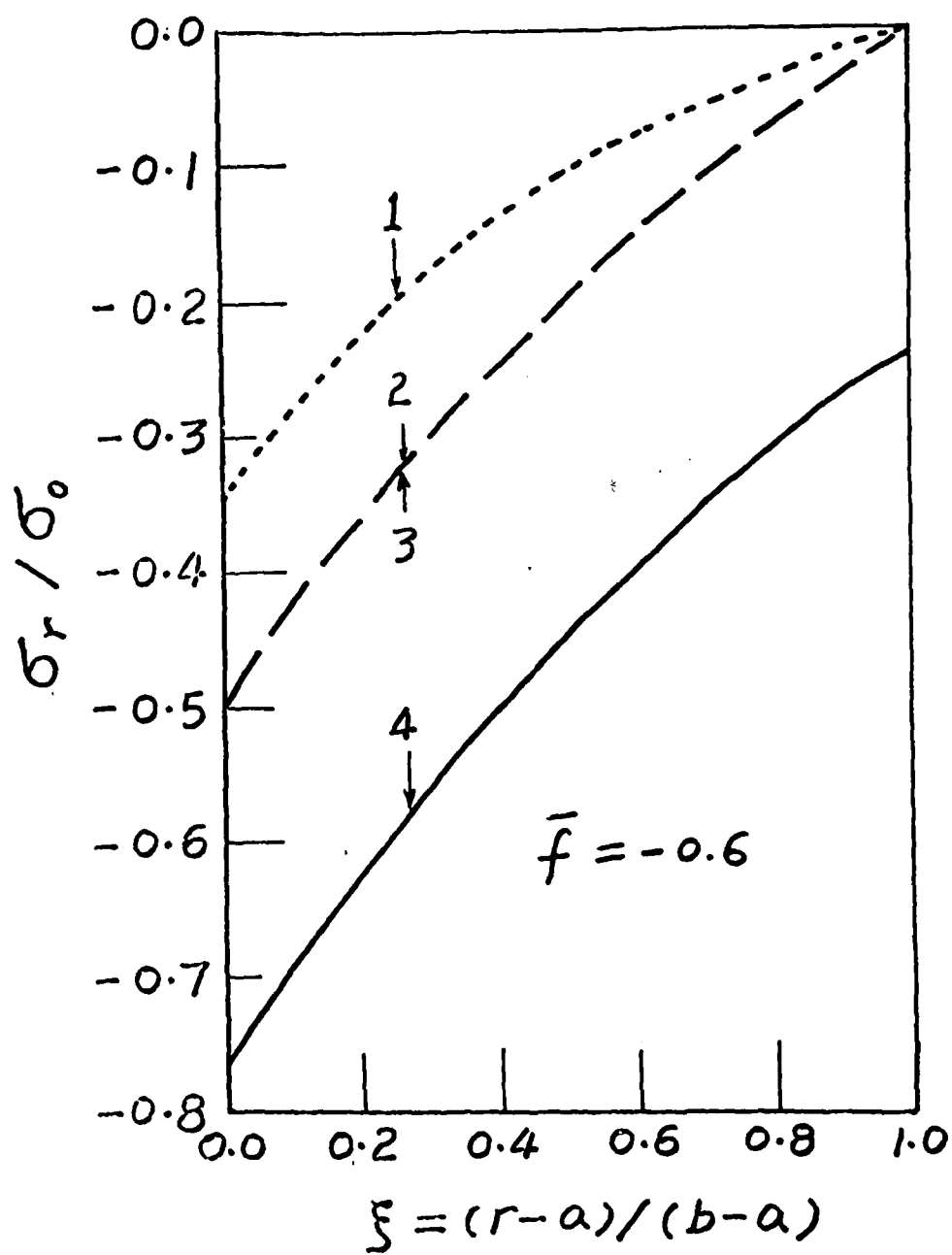


Figure 7. The radial stress distributions during loading with $\bar{f} = -0.6$.

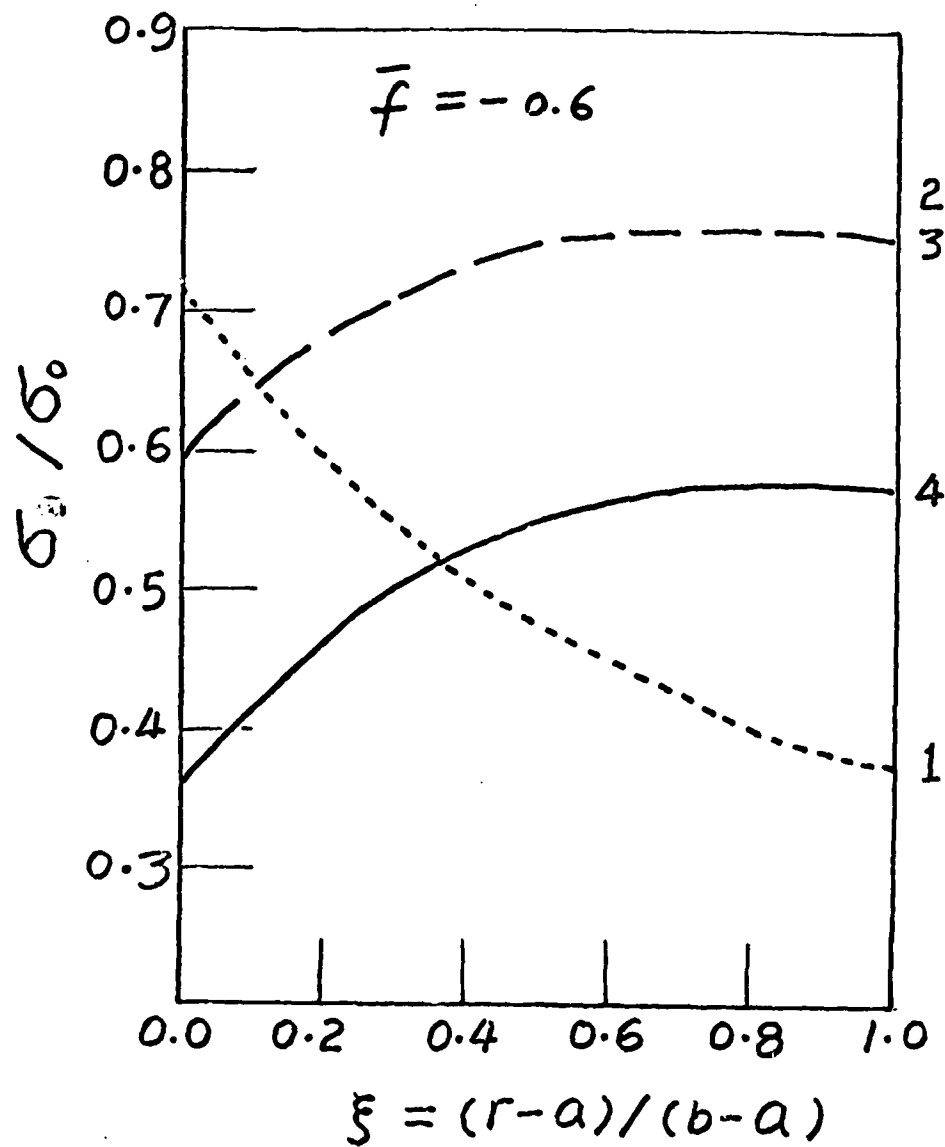


Figure 8. The hoop stress distributions during loading with $\bar{f} = 0.6$.

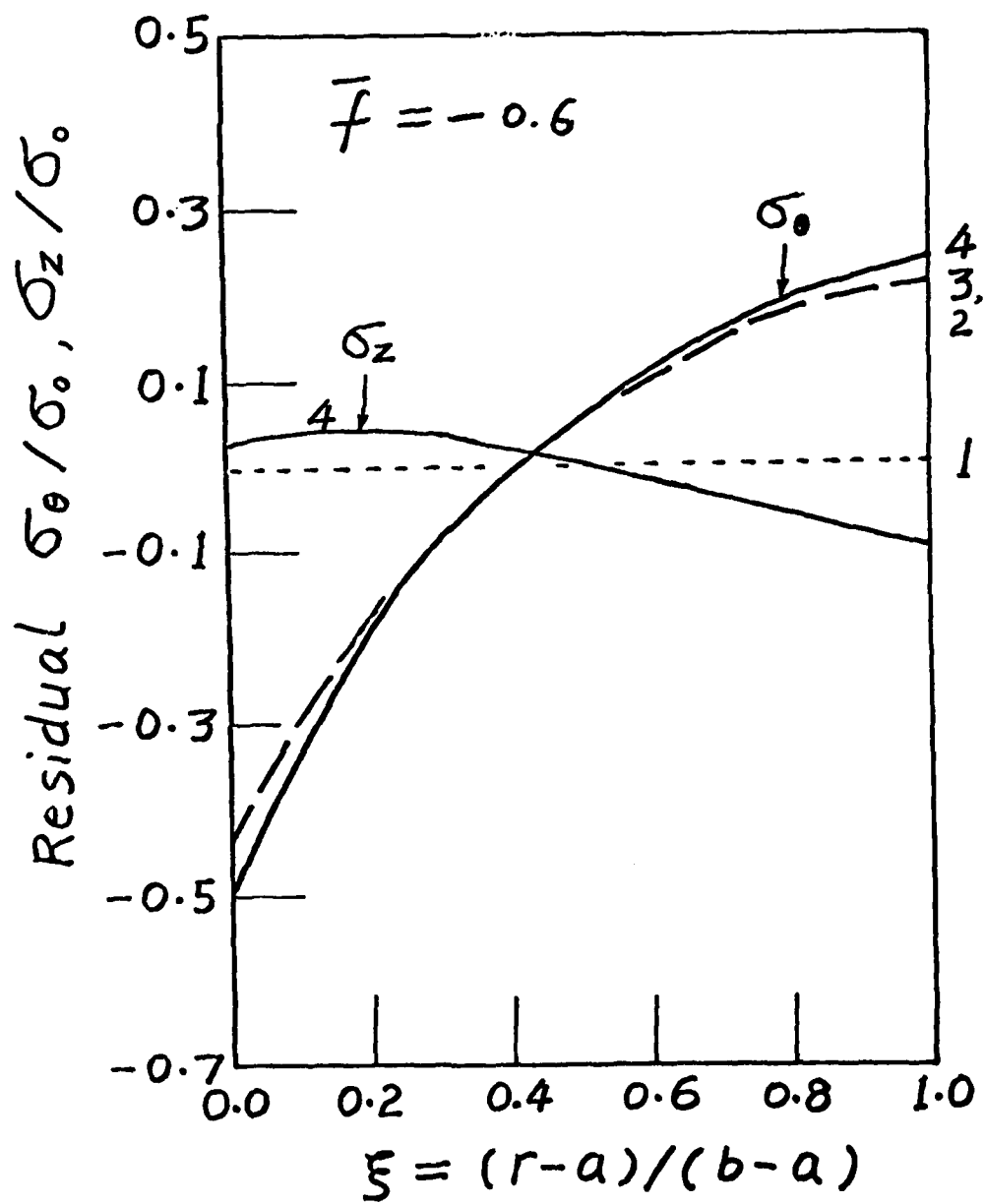


Figure 9. The residual stresses due to complete unloading from different stages with $\bar{f} = -0.6$.

FINITE ELEMENT MODELING OF THE VULNERABILITY OF
U.S. AND FOREIGN LAND MINES TO BLAST LOADS

Frederick H. Gregory
Aaron D. Gupta

U.S. Army Ballistic Research Laboratory
U.S. Army Armament Research and Development Command
Aberdeen Proving Ground, Maryland 21005

ABSTRACT. The structural response of the U.S. M-15 and the Soviet TM-46 land mines to an externally applied pressure wave has been analyzed with the ADINA finite element code. The finite element models of these mines use the axisymmetric two-dimensional mesh configurations with both rigid and non-rigid base support boundary conditions to simulate the soil. An explicit central difference time integration scheme has been used for both analyses.

The mines' steel casings and high explosive filler materials were assumed to have nonlinear constitutive material models. Trapped air inside the mine body was modeled as an assembly of inviscid linear compressible fluid elements. The steel cases were found to be markedly inhomogeneous via 1-D tensile tests of specimens cut from various areas of the mines. These materials were modeled with bilinear stress strain curves, von Mises yield condition, and kinematic hardening rule. Tension cut-off elastic-plastic models of the explosives which employed bulk moduli vs volume strain relations, were derived from Mie-Grüneisen shock wave equations of state. These models allowed tension cut-off planes to form in a direction normal to the principal tensile stress whenever the strain initially exceeded 0.1% in tension.

Solution of these problems in terms of stresses and displacements out to 2 msec of real time response required approximately 4 to 5 hours of cpu time on the CDC 7600 computer for a transient shock load imposed on the top and sides of the mines. Failure of the mine cases was predicted, based on a comparison of the value of the three-dimensional second invariant of plastic strain with that of the one-dimensional value obtained from the tensile tests.

1. INTRODUCTION. This paper describes the response of antitank mines of two different configurations to a transient blast load. The rationale for this analysis is the need to develop a remote, expeditious means of clearing a path through an enemy mine field. A technique of delivering a relatively large transient pressure to the surface of the earth by means of explosives is under development. The object of this study is to determine the extent of structural damage to mine bodies from a given level of blast wave amplitude and shape. The principal kill mechanism is to be a serious distortion or rupture of the mine body rather than fuze initiation or pressure plate removal since the activation mechanisms could be changed easily from one type of mine to another and a sure-kill could not be guaranteed based on a particular mode of actuation.

The mines investigated represent typical antitank mines, both foreign and of U.S. manufacture, which consist basically of round thin metal bodies filled with explosives. These types of antitank mines constitute a large part of the inventory of U.S. and Soviet mines. The components most distinctive are the fuze mechanisms. There are a variety of radically different fuzes for these

mines, different both in mechanical designs and method of activation. Therefore the numerical models adapted for the two mines are representative of a large class of both foreign and U.S. mines.

The paper has four major areas as follows: (a) problem definition, (b) determination of material properties and selection of failure criteria, (c) finite element model description and calculations, and (d) dynamic response prediction of the structural assembly.

2. PROBLEM DEFINITION.

A. TM-46 Antitank Mine Description. The TM-46 land mine has a cylindrical steel body with a primary fuze well in the center of the top and one on the bottom, presumably for antilift or booby trapping purposes. In addition, it has a secondary fuze well in the sidewall underneath the carrying handle. A sectional drawing of the mine is shown in Figure 1. The mine has a nominal diameter of 29.7 cm, height of 7.3 cm, and weighs 8.7 kg with a main charge of 5.7 kg TNT.

The mine body is made of three pieces of sheet steel which are joined at the upper periphery by a 360° crimp. The top cover of the mine body is only .635 mm thick and has three steps. This cover connects to a central circular plate formed by spot-welding of a thick plate to the thin cover section. The intermediate wall is formed from .94 mm thick steel sheet to which a hollow cylindrical piece .56 mm thick is attached to form the centrally located top fuze well. The fuze well contains a 40 g tetryl booster charge for fuze activation.

The lower part of the mine body is formed by a deep drawing operation which results in very inhomogeneous material properties. The central cavity in the main body of the mine is filled with a charge of 5.7 kg TNT explosive. The cavity between the top and intermediate walls is unfilled. However compression of air in this region can contribute to alteration of the response behavior of the mine and subsequent uncrimping of the joint.

The normal method of activation of the fuze is by means of force applied to the pressure cap depressing the fuze and releasing the striker to strike the booster charge in the fuze well. This activates the tetryl booster which in turn detonates the primary TNT charge. The secondary fuze well on the TM-46 mine gives it an antidisturbance capability.

B. M-15 Antitank Mine Description. The M-15 mine has a cylindrical body similar to the TM-46 mine. However there is no intermediate wall or unfilled space in the U.S. mine. The mine has a nominal diameter of 32.13 cm, height of 9.88 cm, and weighs 14.3 kg. The center of the top of the mine has a depressed area which houses the pressure plate assembly. Isometric and side views of the mine are shown in Figure 2.

The mine body is made essentially of two pieces of WD-1010 steel which are joined at the lower periphery by a 360° crimp. The upper part of the mine body is formed by a deep drawing operation which results in very inhomogeneous materials properties as is the case with the Soviet TM-46 mine. The central cavity in the lower half of Figure 2 is filled with 10 kg of composition B explosive.

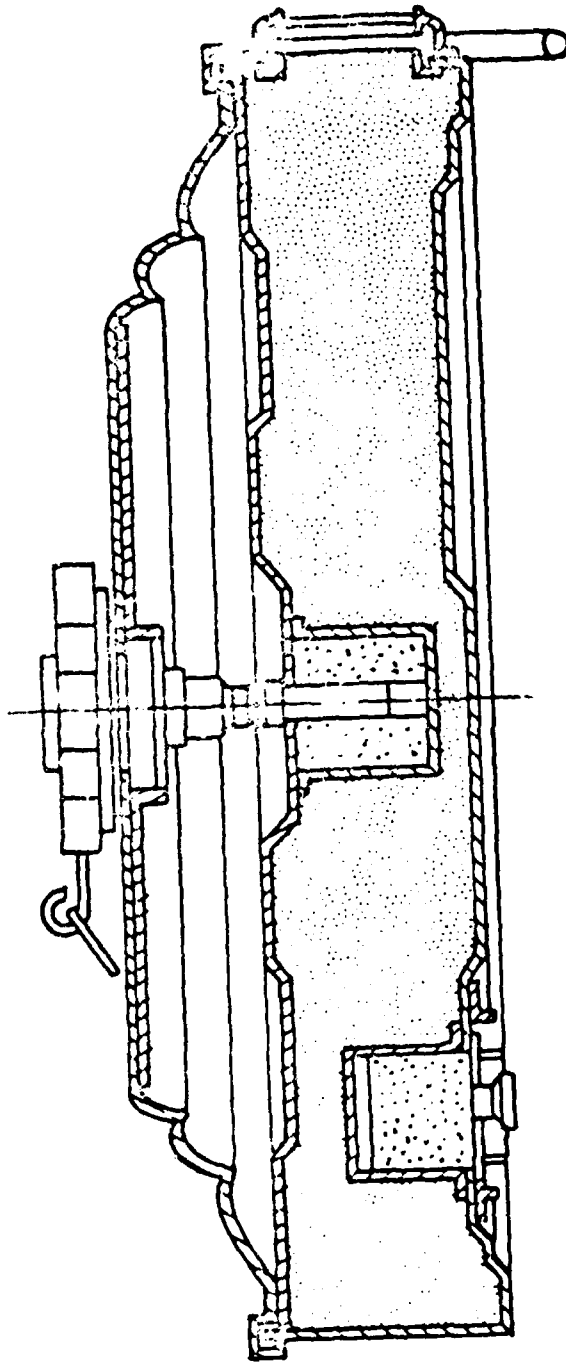


Figure 1. Soviet TM-46 Antitank Mine

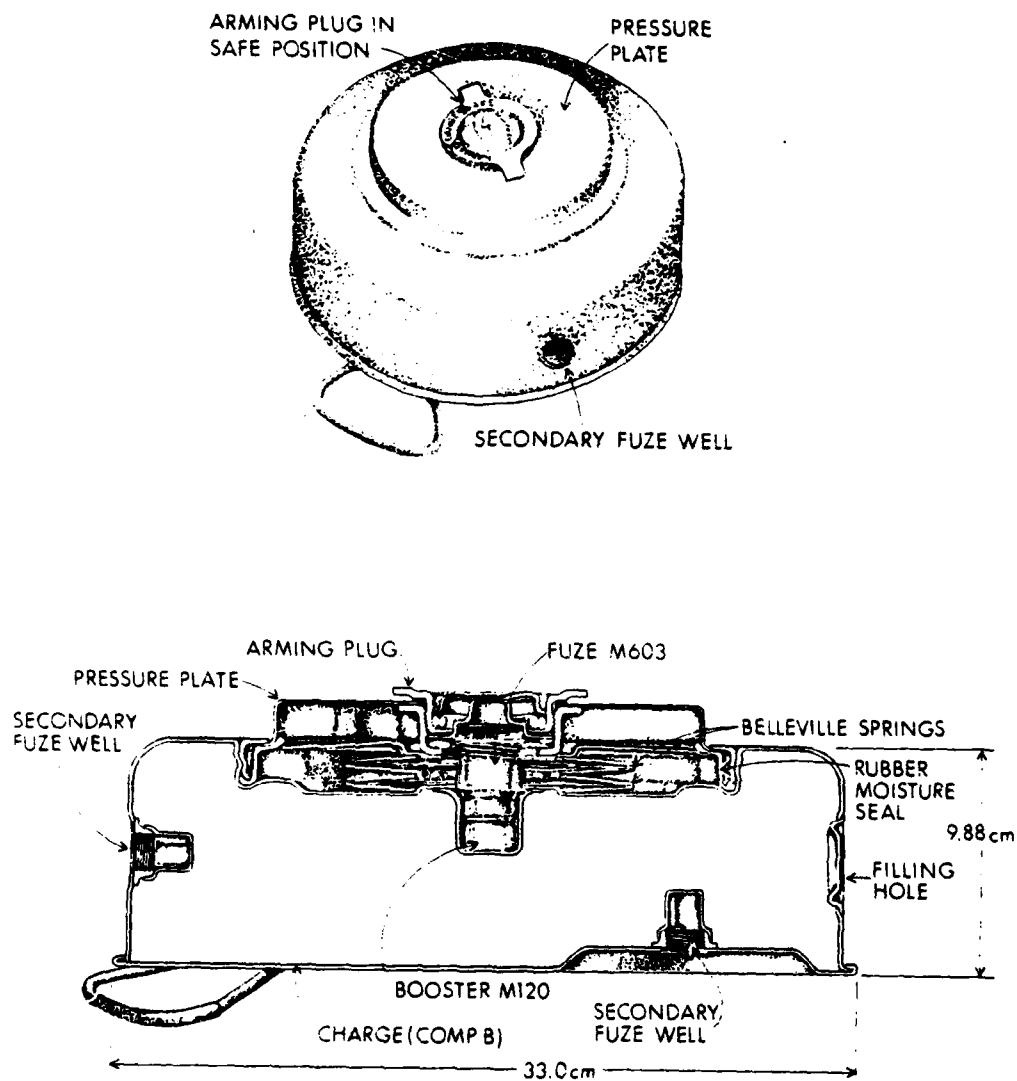


Figure 2. U.S. M-15 Antitank Mine

The fuze is activated by means of force applied to the pressure plate (1250 to 2000 newtons) which in turn is transferred to the belleville springs. At a certain deflection, the belleville springs snap through, driving the firing pin into the detonator. The explosion of the detonator activates the tetryl booster which in turn detonates the primary composition B charge. There are two auxiliary fuze wells on the M-15 mine to allow anti-disturbance capability similar to the soviet mine.

C. Guidelines for the Numerical Model. In keeping with the philosophy of identifying a general failure mechanism independent of some specific design feature, all pressure caps or plates, fuzes and springs were omitted from the finite element model of both mines. This was done in accordance with the previously stated guideline of not identifying failures of the fuze components. The models shown do not include secondary fuzes and filling holes. However the secondary tetryl booster charge is included in the soviet mine to facilitate assessment of the influence of trapped air in the unfilled space below the top wall.

The auxiliary fuze wells were not considered in the current investigation since they make the mine bodies highly susceptible to damage due to stress concentrations near the junction between the body and the fuze. Thus, the simplified model is conservative in terms of blast load required for mine deactivation. Also, inclusion of these unsymmetrically located structures would have necessitated the use of a three-dimensional (3-D) finite element model resulting in significant increase in computing time and costs. The dimples at the base of both mines were eliminated for the same reasons. Because of these simplifications the 2-D axisymmetric models were adequate for dynamic response evaluation.

D. Base Support and Surface Loading. During field emplacement, the mines may be placed on the surface and covered with grass or other materials for concealment. In other cases, the mines may be shallow buried. In either case, the mines will experience transient pressure loading on the top surface due to detonation of a countermine explosive in the vicinity. The base and side boundary conditions were treated in two different ways in the M-15 mine study. It is expected that typical field boundary support conditions would be bracketed by the two extreme conditions simulated. In one case, the base was supported on nonlinear springs, simulating soil. In this case, the mine was simulated as being buried in soil up to its top surface by allowing downward acceleration/movement of the mine based on dynamic properties of the soil medium as described in Reference 1.

The other support condition used for the M-15 and TM-46 mines was a rigid support which closely modeled the experimental conditions described in Reference 2. A roller support condition was used allowing lateral, but no vertical, motion. The indirect loading of the mine through shock waves passing through the soil medium was not modeled. In this rigid support condition, the input shock load is applied to the top and sides of the mine; whereas, in the spring support condition, only the top of the mine was loaded directly.

For structural loading the pressure pulse used in this paper simulated peak pressure and impulse measured from experiments conducted with mine clearance types of explosives in Reference 2. The peak pressure was 13.8 MPa and the

impulse delivered was 6.5 kPa-sec. A decaying exponential function was fitted to these parameters resulting in the following equation

$$P(t) = 13.76 e^{-2117t}. \quad (1)$$

A curve of this function varying in time is shown in Figure 3.

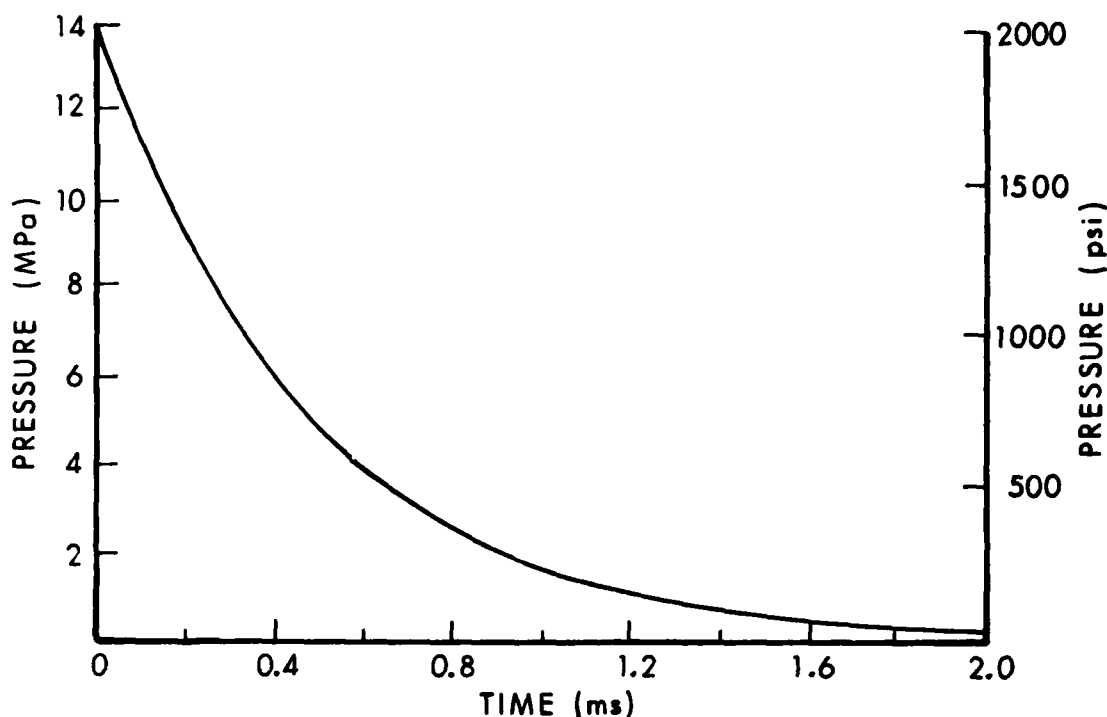
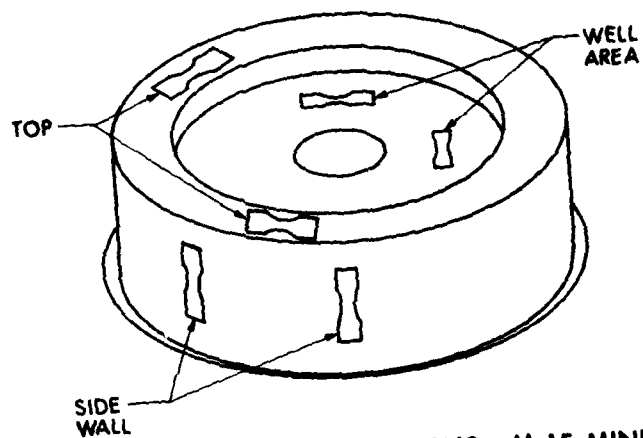


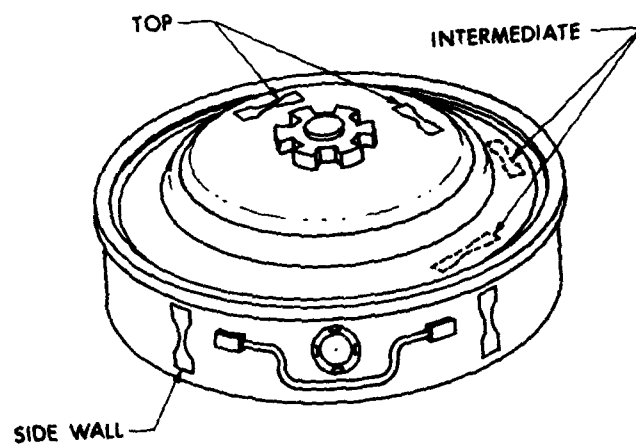
Figure 3. Shock Loading Function for Antitank Mines

3. MATERIAL PROPERTIES AND FAILURE CRITERIA. Material properties were required for the steel jackets, the explosive filler materials, the trapped air, and the soil in which the mine is emplaced. Mechanical properties were measured for the steel jackets by employing uniaxial tensile tests. The data for the explosive and soil were taken from available publications. Failure criteria used for the steel jackets and the filler materials were similar to the formulations in Reference 3.

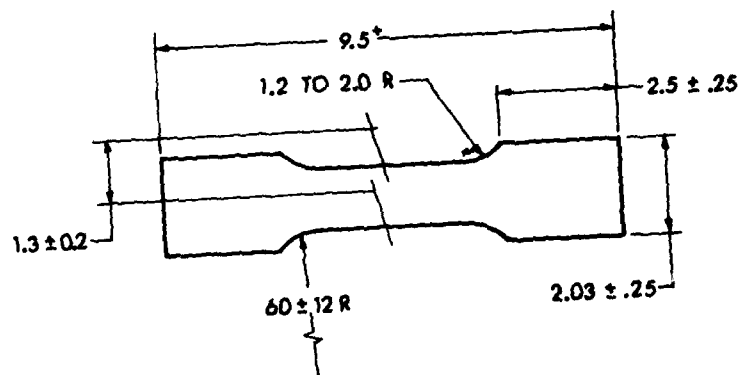
A. Steel Casing. The M-15 jacket is made of a medium strength steel alloy with a density of 7.80 g/cm^3 and a thickness of .94 mm. The TM-46 jacket is made of a low carbon soft magnetic steel equivalent to mild steel. The lower part of the casing was deep drawn, but it retained an equiaxed grain microstructure with isotropic properties. Two tensile specimens were cut from each of the significant surfaces of the mine body. Locations of these specimens are shown in Figure 4(a) and 4(b). The specimens were machined with a large radius on the test section as shown in Figure 4(c). An extensometer and a biaxial strain gage were attached at the location of the minimum width and the specimens were tested in an Instron Testing Machine.



(a) LOCATION OF SPECIMENS, M-15 MINE



(b) LOCATION OF SPECIMENS, TM-46 MINE



(c) PREPARATION OF SPECIMEN DIMENSIONS (cm)

Figure 4. Details of Tensile Specimen Sampling and Preparation

Typical stress-strain curves for the U.S. and the Soviet mine body are shown in Figures 5 and 6 respectively. Evidence of work hardening and residual stress was significant in the Soviet mine due to the forming operation and operating field conditions.

Bilinear approximations to the stress-strain curves obtained by averaging the data for the individual specimens are shown superimposed in Figures 5 and 6. The ADINA (4,5) finite element code used in this analysis has a bilinear, elastic-plastic, von Mises yield condition, kinematic hardening, axisymmetric 2-D element for the steel jacket.

The criterion selected to predict failure of the steel casing material was described in Reference 3 as the value of the second invariant of plastic deviatoric strain at failure, $I_{2f}(\epsilon^P)$, defined as

$$I_{2f}(\epsilon^P) = 1/2 \epsilon_{ij}^P \epsilon_{ij}^P, \quad (2)$$

where the strains indicated are to be the strains at failure. In the uniaxial tension test where the load is applied in the Z-direction, we have,

$$I_{2f}(\epsilon_{1-D}^P) = 3/4 (\epsilon_{ZZ}^P)^2. \quad (3)$$

B. Characterization of Explosives. There are two types of explosives employed in the TM-46 mine, i.e., TNT as the main charge and tetryl as the fuzewell charge. For the U.S. M-15 mine, composition B-3 explosive consisting of 60% RDX and 40% TNT is used in cast form as the main charge.

After surveying the available material properties of explosives and the various 2-D axisymmetric materials models in the ADINA code, it was decided that the curve description material model (see Section XII p. 17-22, Ref. 4) was the appropriate model to use. This model requires tables of loading and unloading bulk moduli and shear moduli versus volume strain.

A relationship between the volume strain and the bulk modulus obtained from the Mie-Grüneisen equation of state (Reference 1,6) is given as

$$\kappa = \frac{\Gamma(\Gamma + 1)(A\mu^2 + B\mu^3 + C\mu^4)}{2 - \mu\Gamma} + A + A'\mu + B'\mu^2 + C'\mu^3 \quad (4)$$

where

- κ = the loading bulk modulus
- Γ = the Grüneisen coefficient
- A, B, C = the coefficients appearing in the Grüneisen equation of state in terms of μ
- $A' = A(\Gamma+1) + 2B$
- $B' = B(\Gamma+2) + 3C$
- $C' = C(\Gamma+3)$

PRESSURE PLATE WELL TENSILE TESTS

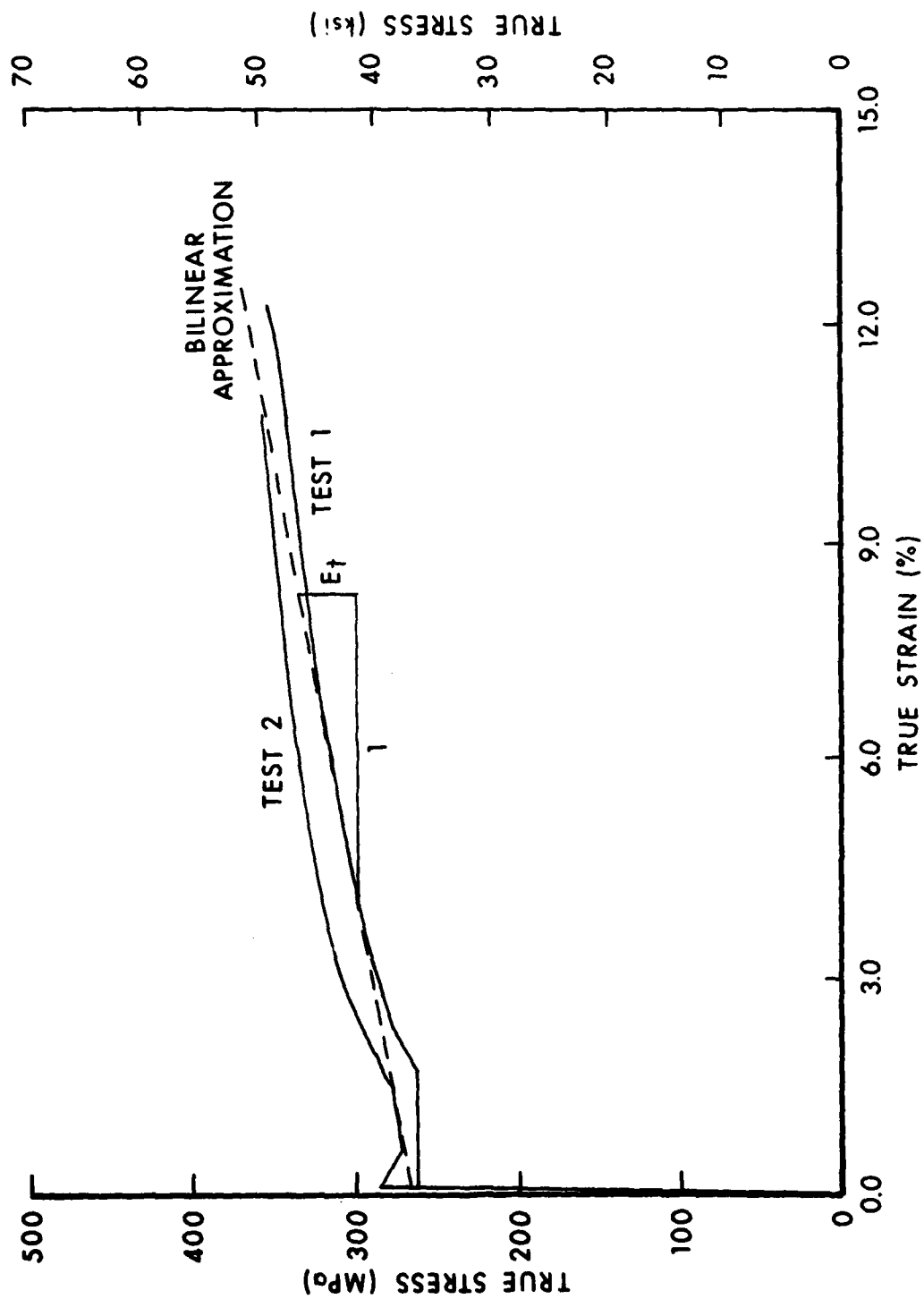


Figure 5. Stress-Strain Curves for the Pressure Plate Well Specimens for the M-15 Mine

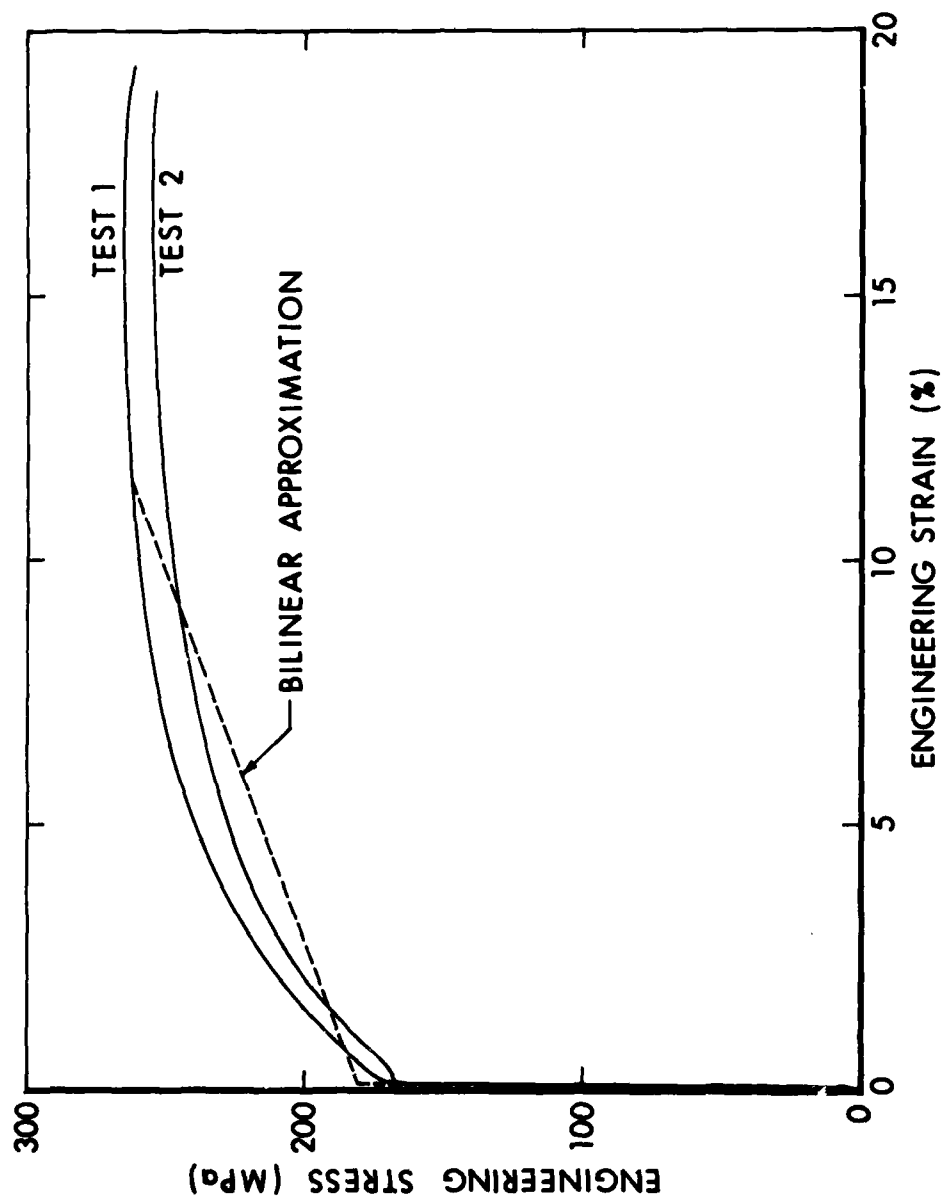


Figure 6. Stress-Strain Curves for the Top Cover Plate Specimens for the TM-46 Mine

$$\mu = \epsilon_v / (1 - \epsilon_v)$$

$$\epsilon_v = (V_0 - V) / V_0, \text{ volume strain taken positive in compression}$$

$$V_0 = 1/\rho_0 = \text{specific volume at normal conditions.}$$

The values for the material constants of the explosives used are shown in Table 1.

TABLE 1. MATERIAL CONSTANTS FOR EXPLOSIVES AND SOIL

Type	ρ_0 (g/cm ³)	Γ	A (Gpa)	B (Gpa)	C (Gpa)	ν
Comp B-3	1.68	.947	13.5	9.5	100.6	.29
TNT	1.614	.737	10.367	9.101	138.33	.3
Tetryl	1.70	1.6	10.498	17.8	20.6	.3
Wet Tuff	2.0	1.5	21.77	32.5	18.33	—

Note that when $\epsilon_v = 0$, $\mu = 0$, $\kappa_0 = A$ and $V = V_0$. Also, in the Grüneisen EOS, at $\epsilon_v = 0$, we take the pressure and internal energy to be zero.

Because no data were available to relate the unloading bulk modulus to the volumetric strain, the same values of the bulk modulus for unloading as for loading were used for all explosives. The loading shear modulus, G_ℓ , was obtained from the loading bulk modulus, κ_ℓ , by use of the relationship,

$$G_\ell = \frac{3\kappa_\ell(1-2\nu)}{2(1+\nu)} \quad (5)$$

Figures 7-9 show the graphical relationships of the three explosives represented by Equations (4) and (5). Table 2 gives the values of the two moduli as they were used in the ADINA program. ADINA uses linear interpolation between discrete points.

The tensile volumetric strain at failure for the composition B-3 explosive is given in Reference 6 as -0.1 per cent. This criterion was used in calculations for all explosives in this investigation. The technique used in the ADINA code to apply this failure criterion is by the artifice of superimposing on the applied load-induced strains, an in-situ gravity pressure sufficient to cause a hydrostatic compression equal in magnitude to the tensile failure. Then, when the total strain becomes negative, a tension cut-off plane is assumed to form normal to the principal strain. The normal and shear stiffnesses across this plane are reduced by a factor determined by an input value. One or two additional planes orthogonal to existing tension cut-off plane(s) are allowed to form if the strain criterion is met. The planes become inactive if compression again develops in the direction normal to it.

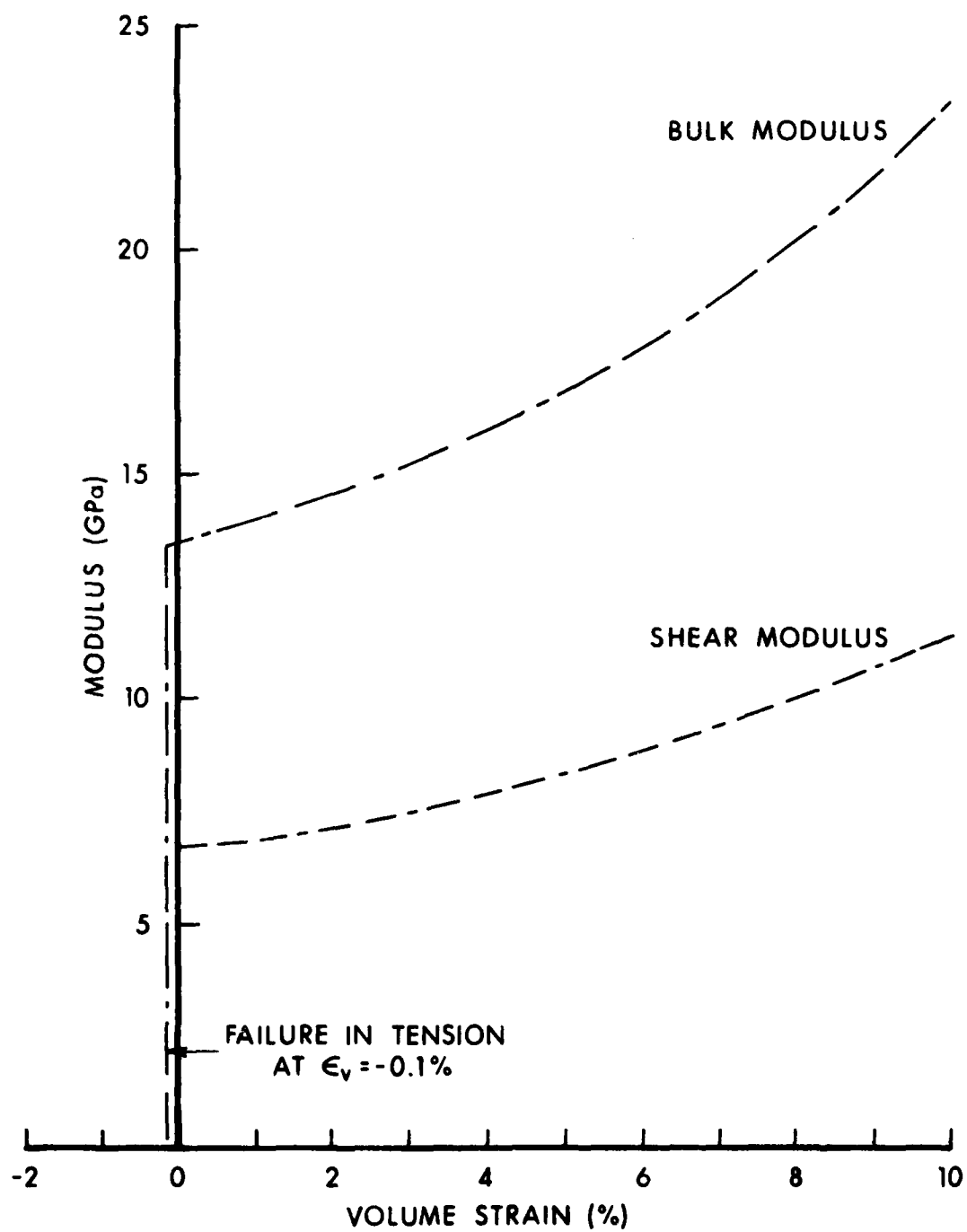


Figure 7. Bulk and Shear Moduli vs Volume Strain for Composition B-3 Explosive of the M-15 Mine

AD-A118 920

ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82
ARO-82-3

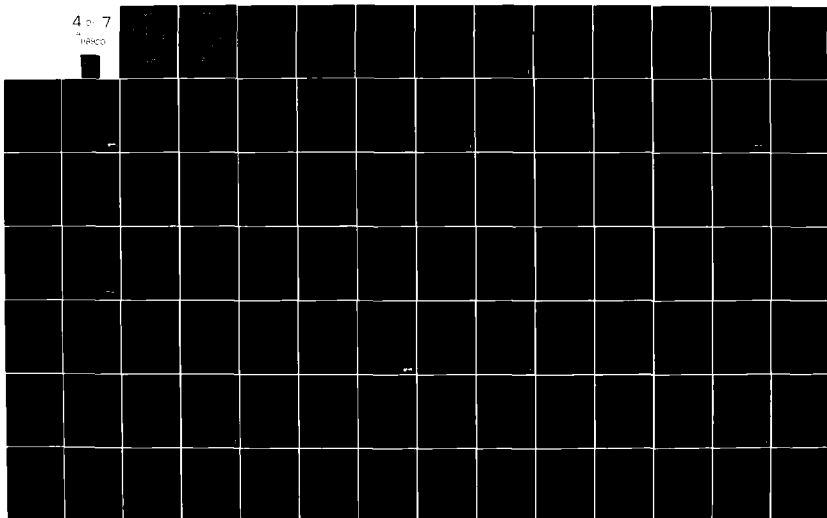
F/O 12/1

NL

UNCLASSIFIED

4-7

PERCO



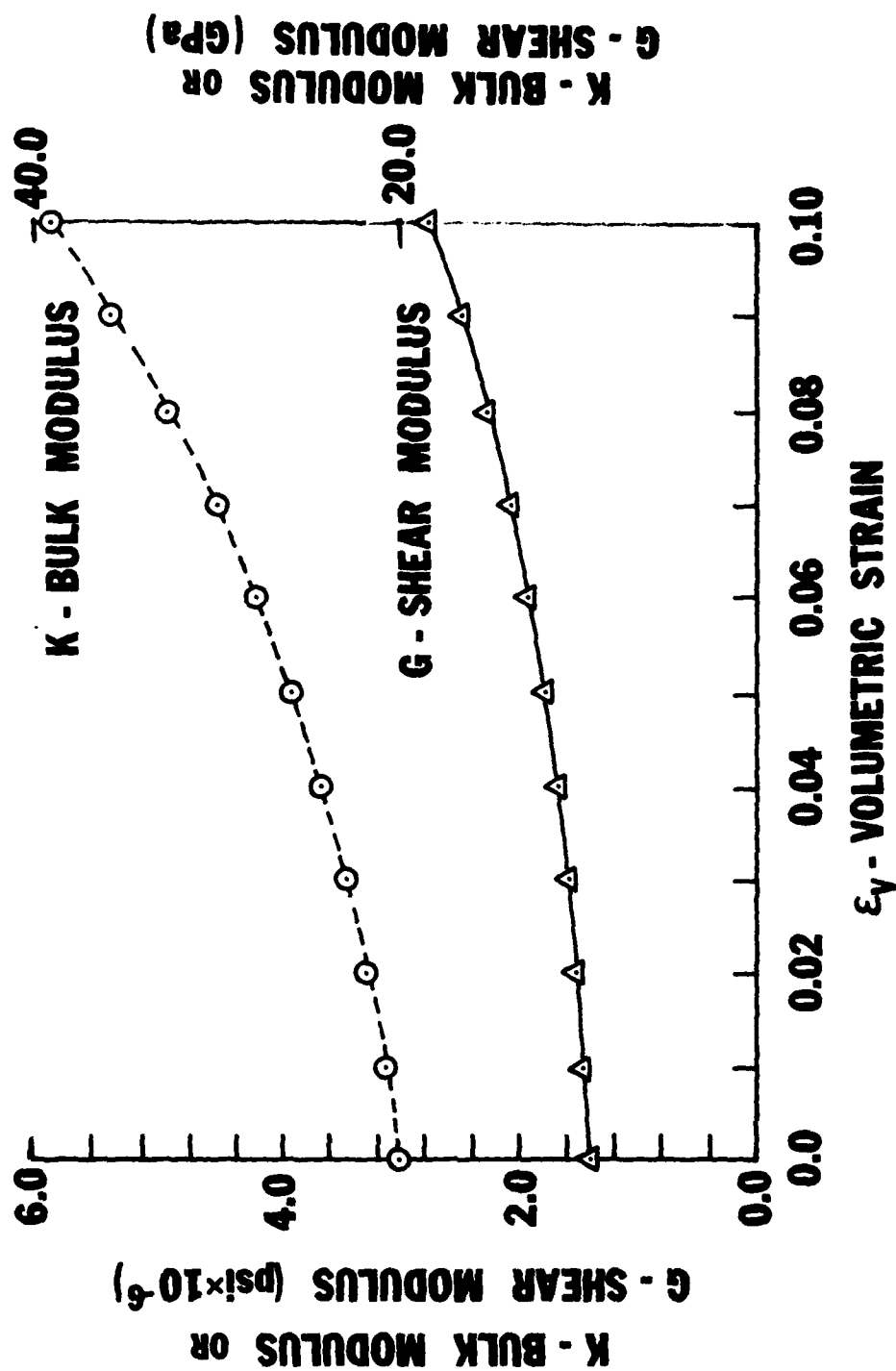


Figure 8. Bulk and Shear Moduli vs Volume Strain for TNT Explosive of the TM-46 Mine

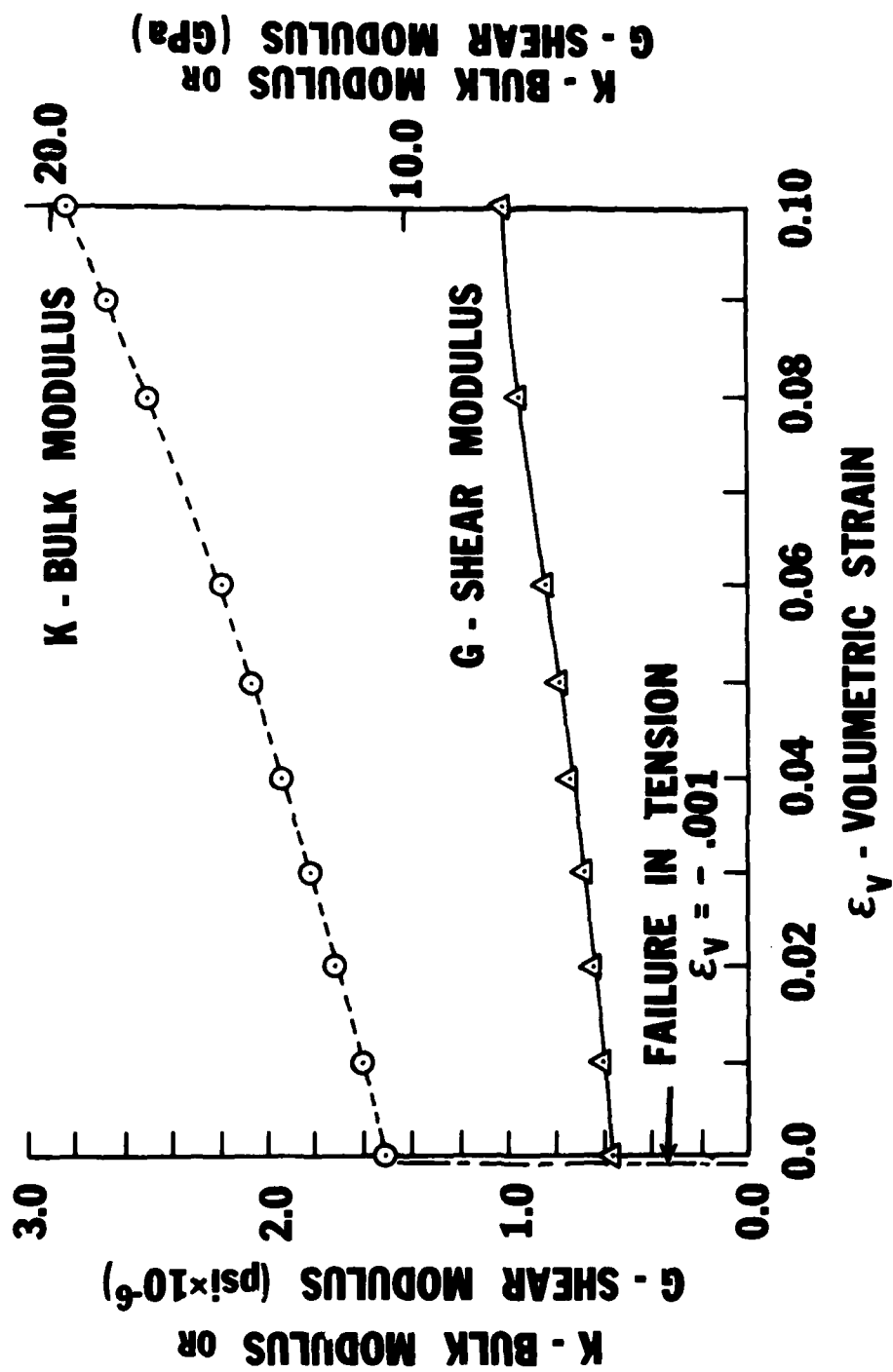


Figure 9. Bulk and Shear Moduli vs Volume Strain for Teteryl Fuze Well Charge of the TM-46 Mine

TABLE 2. ADINA INPUT VALUES FOR BULK AND SHEAR MODULI FOR FILLER MATERIALS

COMPOSITION B-3 EXPLOSIVE

Point No.	ϵ_v (%)	κ_l (GPa)	κ_u (GPa)	G_l (GPa)
1	0	13.52	13.52	6.60
2	1.0	14.00	14.00	6.84
3	2.5	14.91	14.91	7.28
4	3.75	15.83	15.83	7.73
5	5.0	16.92	16.92	8.26
6	10.0	23.36	23.36	11.41

TNT EXPLOSIVE

1	0	21.72	21.72	10.62
2	1.0	23.03	23.03	11.24
3	3.0	25.65	25.65	12.55
4	5.0	28.68	28.68	14.01
5	9.0	35.85	35.85	17.51
6	11.0	40.20	40.20	19.65

TETRYL FILLER

1	0	10.5	10.5	4.03
2	1.0	11.15	11.15	4.27
3	3.0	12.59	12.59	4.83
4	5.0	14.24	14.24	5.46
5	8.0	17.2	17.2	6.60
6	10.0	19.56	19.56	7.50

The pseudo-hydrostatic pre-strain is applied by positioning the vertical coordinate (Z-coordinate) at the proper negative value. The hydrostatic pressure applied at an element integration point is given for an element, j, by

$$P_j = - \rho_e \sum_{i=1}^N h_{ij} Z_{ij} \quad (6)$$

where

ρ_e is the density of the overburden

h_{ij} is the shape function for node i of element j

Z_{ij} is the vertical coordinate for node i in element j.

The position of the system vertical coordinate can be obtained from the equation,

$$z_{ave} = \frac{\kappa_o \epsilon_v^f}{g \rho_e}$$

where

κ_o is the initial bulk loading modulus,

ϵ_v^f is the volumetric failure strain, negative in tension,

g is the acceleration due to gravity.

C. Soil Simulation. For the structural response calculations of the shallow buried M-15 mine, only the top of the mine was exposed to blast pressure while the remainder was assumed to be embedded in soil. An implicit modeling technique was employed whereby nodal tie elements were used to model the base support as nonlinear springs. No simulation of the soil was necessary for the rigid support calculations.

Three different types of nodal tie elements were available in the Ballistic Research Laboratory version of the ADINA code. The particular type chosen is the boundary type element defined by one node only and is capable of three translational and three rotational degrees of freedom. In the M-15 mine, the elements along the base of the mine were used to transmit a vertical force (F_z), while those along the side exerted a horizontal force (F_y).

Due to the large variety of soils in which mines would be emplaced, it is possible only to select a soil simulation model which would be representative of some subclass of soils. Thus, a typical load deflection curve (Reference 7) was selected to define the nodal tie element properties. The average load-deflection for slowly varying loads in the elastic loading range from Reference 7 is .0815 MPa/cm. To account for the dynamic response of soil at the base of the mine, a nonlinear quadratic component was added to the force deflection property.

For the support along the vertical sides of the mine, a linear spring force was used due to consideration of small lateral movement. The linear nodal tie element stiffness values along the vertical side are proportional to the height of the particular element onto which the nodal tie boundary element is attached. Similarly, the nonlinear stiffness values for springs at the base in the ADINA input data are adjusted by a factor proportional to the annular sector of π radians and a radial extent appropriate for the particular nodal tie element. For soil modeling of the TM-46 mine in the ADINA code, an explicit technique using two layers of compressible soil elements surrounding the mine will be employed.

D. Simulation of Void in TM-46 Mine. The TM-46 mine has a cavity between the upper pressure plate and the middle plate covering the primary charge. This cavity has air in it which would transfer some load to the middle plate as the volume of the cavity is decreased. An attempt was made to model the air with 2-D axisymmetric fluid elements composed of an inviscid linear compressible material. A constant bulk modulus was used in lieu of a pressure

dependent bulk modulus due to lack of available data for air. However, the primary difficulty with this model was that there was nothing in the model to prevent the upper plate from penetrating the middle plate as the deformation progressed.

Since the air was judged to apply only a minimal restraint on the motion of the upper plate and due to the need to prevent the two plates from passing through one another, a different model has been adopted. The model consists of axial truss elements connecting the two circular plates. The material model for the trusses is nonlinear and develops only a small force up until the axial strain in trusses approaches -1. At this strain, a large stiffness is specified to simulate contact between the two plates. Constraints are applied to the upper end of the trusses to insure that its radial coordinate is the same as the radial coordinate of its lower end. Also, the axial coordinate of upper end is constrained to translate with the upper plate.

4. FINITE ELEMENT MODEL DESCRIPTION AND CALCULATIONS. The two mines were modeled as axisymmetric 2-D structures using the ADINA finite element code. The steel components were modeled with six-node elements including mid-side nodes on the plate surface. The explosive components were modeled with four-node QUAD elements except where they interfaced the steel jacket, in which case a mid-side node was included on the interface edge.

The time step used for the calculations was determined from the Courant stability condition

$$\Delta t = \frac{\Delta t_{crit}}{n} = \frac{\Delta l}{n \sqrt{E/\rho}}, \quad (7)$$

where

Δt_{crit} is the minimum Courant stability time step,

Δl is the distance between the two closest nodes in the system,

E is the Young's modulus for the stiffest material,

ρ is the density of the material,

and n is the number of time steps which we wish to represent the shock wave in passing through the distance Δl .

The value of Δt_{crit} was approximately 200 nanoseconds, for both the TM-46 and M-15 mines. A value of n of four was used, so that the time step for the central difference explicit time integration method was 50 nanoseconds.

A. M-15 Mine Calculations. As indicated previously in Section 2D, two different boundary conditions were used in modeling the M-15 mine. The primary difference between the two calculations was in the base support condition. One used a nonlinear spring support and the other used a rigid vertical base support. The mesh configuration for both M-15 models is shown in Figure 10.

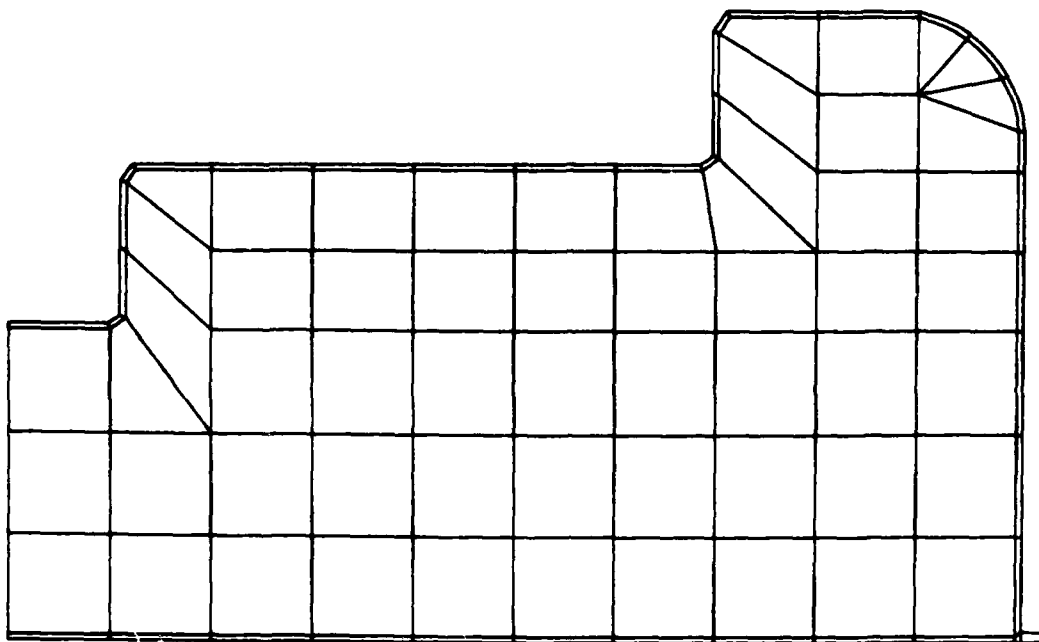


Figure 10. Finite Element Mesh for the M-15 Mine

Eigenfrequencies were generated and the associated mode shapes were plotted via the ADINA post-processor, PLOT3D [8]. The natural frequencies are important for estimating the rate of response of a structure. For similar loadings, the higher the natural frequencies of a structure, the faster the structure will respond. In addition, rapid response causes higher strain rates to be effected. This is significant for strain rate sensitive materials such as mild steel which both of the subject mines embody. However, strain rate sensitivity was not modeled in these calculations. The mode shapes associated with the lower eigenfrequencies often give a good indication of the deformed shape which will result from the application of typical loads. This was especially evident in the deformation of the TM-46 mine. The lower eigenfrequencies and periods for the M-15 mine are given in Table 3.

TABLE 3. EIGENFREQUENCIES AND PERIODS FOR THE M-15 MINE

Spring Supported Mine		Rigidly Supported Mine	
Frequency (cps)	Period (sec)	Frequency (cps)	Period (sec)
36*	2.744×10^{-2}	6426	1.556×10^{-4}
3636	2.750×10^{-4}	7899	1.266×10^{-4}
6710	1.490×10^{-4}	9685	1.032×10^{-4}
8531	1.172×10^{-4}	12186	8.205×10^{-5}

*Rigid body mode.

B. TM-46 Mine Calculations. The ADINA calculations for the TM-46 mine have not been completed. However, some of the salient features of the model have been developed from progress made in studies of the mine thus far. A drawing of the current mesh configuration is shown in Figure 11.

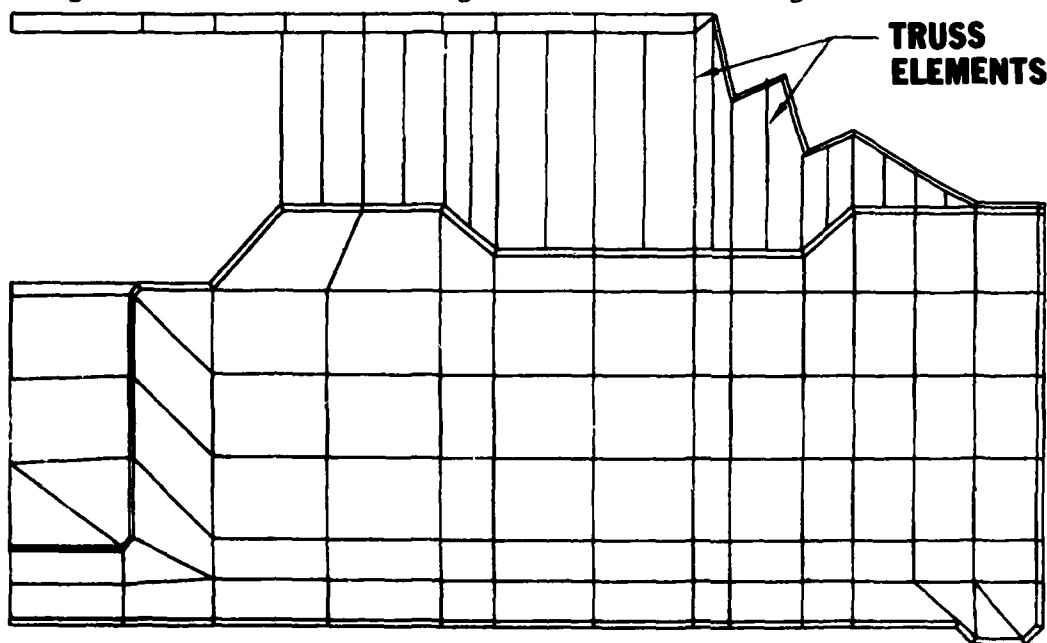


Figure 11. Finite Element Mesh for the TM-46 Mine

From our experience with the M-15 mine, we expected significantly different materials properties in the outer steel jacket of the TM-46 mine. Measurements showed that this was, indeed, the case. Several different sets of materials properties were used to model the various steel components of the mine.

It was evident from the first that two particular difficulties would be encountered in modeling the TM-46 mine. First, the difference in stiffness between the steel plates and the air filled region leads to numerical problems.

The collapse of the air filled region leads to the impact of the upper plate on the middle plate. This phenomenon needs to be modeled rather carefully. Second, the thin stepped top cover shown in the upper right part of Figure 11 leads to a very inefficient load transfer from the top cover to the main mine body. On the other hand, any viable failure mechanism for the mine must inevitably involve a failure of the main mine body.

Since the ADINA code does not currently have a contact element to sense when the top cover plate and middle plate impact, we have used nonlinear truss elements to approximate the interaction of the two plates. This approach was described in Section 3D.

Eigenfrequencies and mode shapes were also obtained for the TM-46 mine model. The eigenfrequencies and associated periods for the lower modes are given in Table 4.

TABLE 4. EIGENFREQUENCIES AND PERIODS FOR THE TM-46 MINE

Frequency (cps)	Period (sec)
3041	3.288×10^{-4}
10466	9.555×10^{-5}
17068	5.859×10^{-5}
31071	3.218×10^{-5}

All calculations described herein used the total Lagrangian formulation with a lumped mass matrix with the exception of the nodal tie and truss elements. The formulations used for these were material nonlinearity only and updated Lagrangian analysis procedure, respectively.

5. DYNAMIC RESPONSE PREDICTIONS. Several modifications to the ADINA program were made to assist us in interpreting the response predictions. These are described fully in Reference 1. A summary of these modifications will be given here. Due to the very large amount of stress-strain data available from the ADINA results, some means of selectively extracting significant parts of the results was desired. Since the component which involved the most credible failure mechanisms was the steel jacket, we focused our attention on it. The modifications were made in two different areas. First, routines were written to monitor the extreme (maximum/minimum) stresses and strains in the steel components. Information on the location, time, and value of these extreme stresses and strains were saved and printed at intervals during the calculation. Second, routines were written to calculate and monitor the second invariant of plastic strain. The value of this quantity was compared to an input value in order to predict failure of the steel jacket. Tables of the maximum value of this quantity were stored and printed at preselected intervals.

In addition to the above modifications to ADINA, one further modification was necessary to successfully obtain the solution to such long response times using the explicit time integration scheme. In the standard ADINA program, whenever plasticity occurs in the kinematic hardening model for a solid element,

a linearized correction is applied to bring the stress tensor back to the von Mises yield surface. Because the linearized correction leaves the stress at a position in stress space along a tangent to the convex yield surface, the resulting stress will be slightly outside the yield surface. An accumulation of error results from this linearization and after many time steps causes imaginary roots to be obtained in solving for the stress correction. It was necessary to include the quadratic stress correction to avoid this problem. The details of this modification are described in References 1 and 9.

A. M-15 Mine Dynamic Response. The calculations for both base support conditions were run to 2.0 milliseconds of response time. This amount of response time corresponds to seven and twelve times the period of the fundamental distortional eigenmode for the spring supported and rigidly supported mine, respectively (see Table 3).

The first predicted failure of the spring supported mine (simulating a mine buried in soil) occurred at 0.67 milliseconds. The failures occurred in the fuze well in the center of the mine as shown in Figure 12. Other areas of the

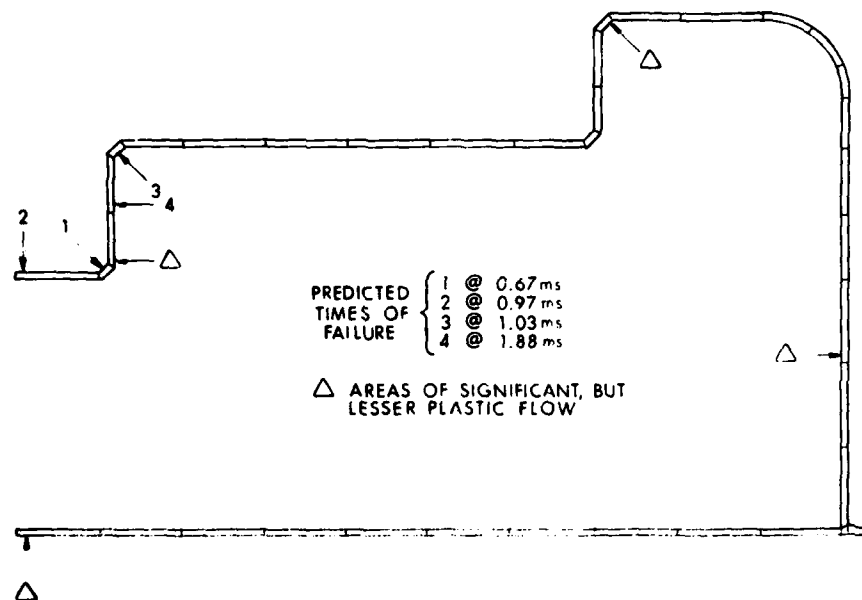


Figure 12. Predicted Failures of the M-15 Mine with Nonlinear Spring Supported Base

mine casing had severe plastic flow as indicated in Figure 12; however, the second invariant of plastic strain did not reach the failure value. Experiments described in Reference 2 showed a similar behavior. Fuze wells were torn from the steel jacket and explosive material was ejected from the inside cavity. The deformed shape of the mine at the time of the first failure of the casing is shown in Figure 13. In making this plot, rigid body motion of the mine on the spring support was subtracted and the resulting displacements were magnified by

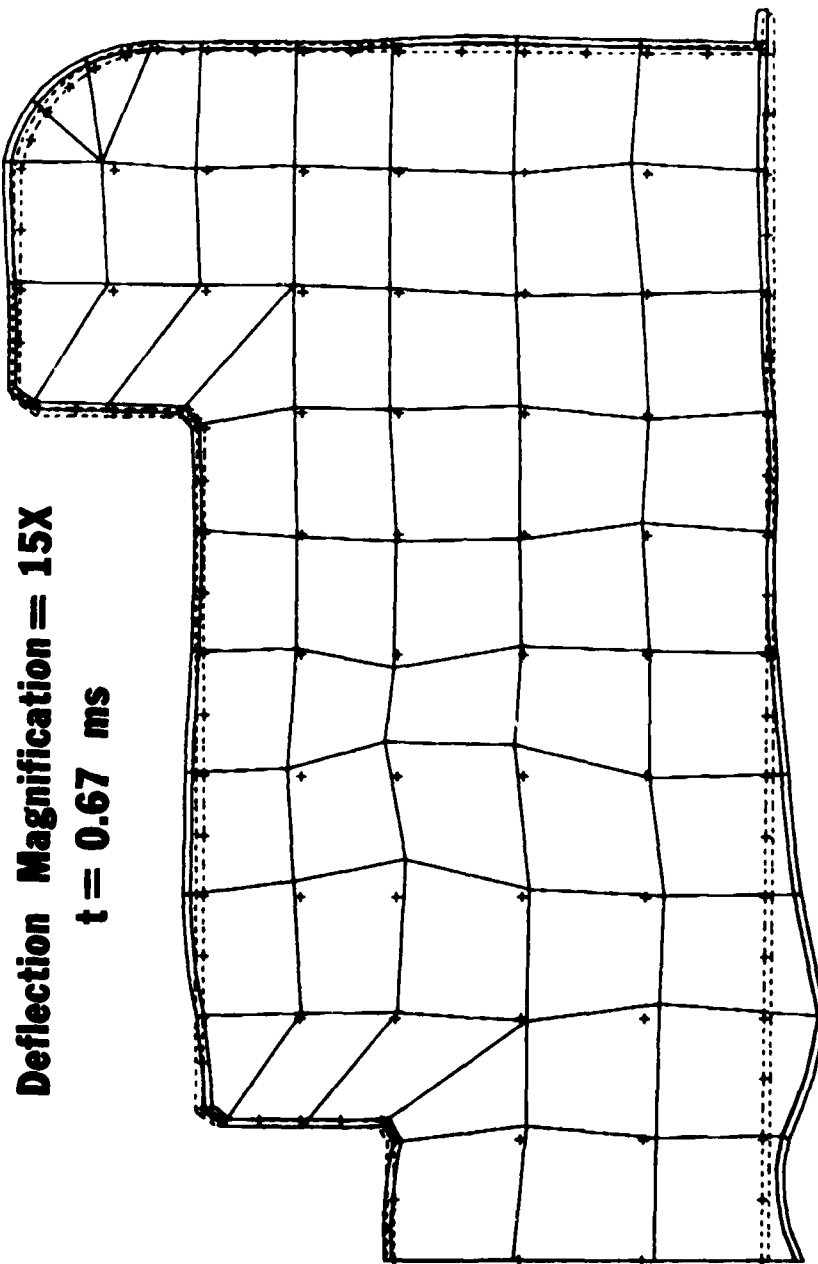


Figure 13. Deformed Shape of the M-15 Mine on Nonlinear Spring Support at the Time of the First Predicted Failure

15 to make the deformation pattern more graphic. The dotted lines in Figure 13 indicate the undeformed shape of the metal casing and the crosses indicate the original nodal position. Contour plots of radial and hoop stress showed high stress gradients in the neighborhood of the predicted rupture points. Extensive cracking of the explosive filler material occurred in these calculations according to the 0.1% tensile strain failure criterion. It would be reasonable to assume that some of the crushed explosive filler material would be expelled through any ruptures which occurred in the casing.

The finite element calculations of the M-15 mine on rigid roller support were also carried out to 2.0 milliseconds. This configuration is a much more highly constrained structure than was the previous case. This fact is reflected in the higher natural frequencies (Tables 3,4). The failures predicted for this configuration occurred in the same general area, the central fuze well. However, the times required for failure to occur were much shorter than those for the spring supported mine as one might expect. The failure of this mine was predicted at two locations at times of 0.255 and 0.609 milliseconds. The first failure of the fuze well occurred in the center on the initial downward compression phase. In the spring supported response, the first failure occurred in a rebound motion of the fuze well. These responses are described more fully in Reference 1.

B. TM-46 Mine Dynamic Response. The deformation of the TM-46, which we have modeled with the ADINA program has been nearly all in the area of the top cover plate. One of the chief difficulties encountered has been in trying to provide the appropriate model for the interaction of the top cover plate on the middle plate. We have used the improvisation of nonlinear truss elements (see Figure 11 and Section 4B) to simulate the impact of these two components. A typical response of the system at an early time is shown in Figure 14. As was the case with Figure 13, the dotted lines represent the undeformed or original configuration before imposition of the blast load. The vertical lines between the top cover plate and the middle plate represent the nonlinear truss elements. Currently, the calculation has not proceeded to the point where any failure of the main mine body can occur. The model of this mine is still evolving.

6. CONCLUSIONS. The explicit time integration method gave the most accurate results for the shock loaded mines. This statement is based on the smoothness of the stresses and strains as a function of time. We found that second order corrections to assure that the stress state is on the yield surface during plastic flow are required to keep the calculational procedure from failing.

The parts of the outer steel jacket of the M-15 and TM-46 mines which are work hardened in the deep drawing metal forming operation have significantly varying materials properties. These variations in stress-strain relations must be measured and modeled carefully since they have a direct influence on mine failure under blast loads.

The soil medium supporting the mine and the nature of the loading of the sidewall have a significant influence on the resulting response. It is recommended that the soil medium be included explicitly in any future studies.

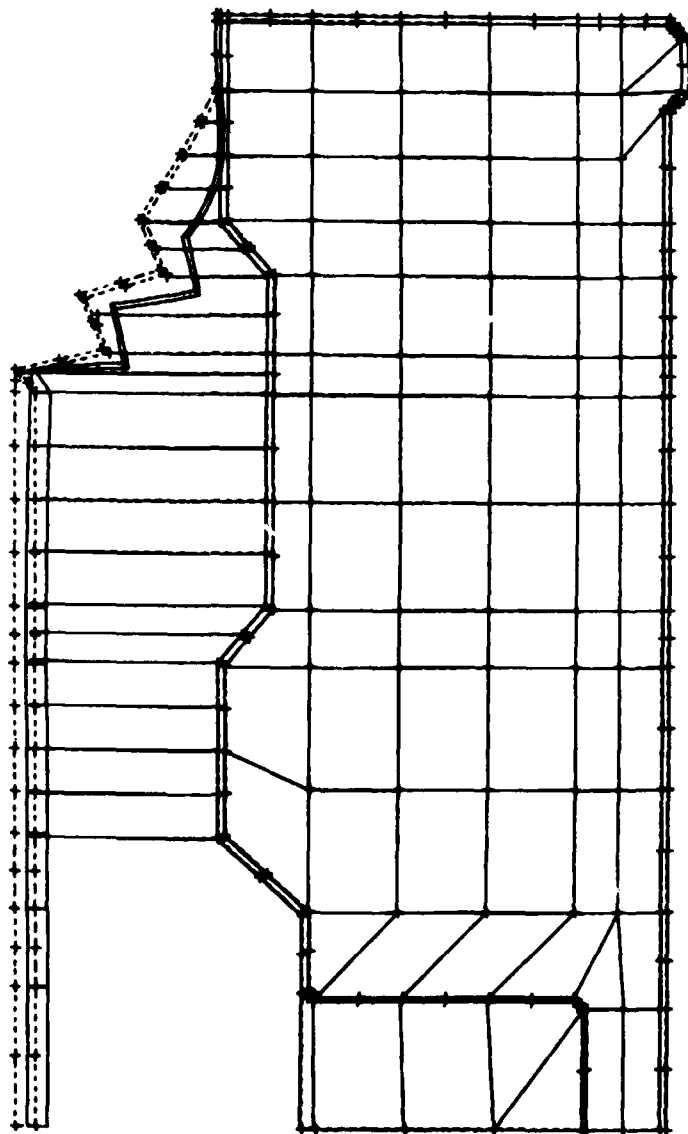


Figure 14. Deformed Shape of the TM-46 Mine on Rigid Support at 75 Microseconds

Attenuation of the shock in the soil in the neighborhood of the mine sidewall should be investigated.

Failure of the M-15 mine occurred in the area of the central fuze cavity when subjected to a 13.8 MPa peak pressure, 6.5 kPa-sec impulse level blast load in both the rigid support and soil support simulations. This agrees with experimental tests in which catastrophic failure of the metal casing occurred, as well as ejection of secondary fuze wells.

The initial deformation of the Soviet TM-46 mine was limited to the response of the top cover plate. The finite element modeling with the ADINA program has presented some difficulty in describing accurately the impact of the top plate on the middle plate (plate containing the primary TNT charge). This study is still in progress.

REFERENCES.

1. Frederick H. Gregory, "Failure of the M-15 Antitank Mine Due to Blast Loads," BRL Report in publication.
2. Allen J. Tulis et al (IIT Research Institute) and David C. Heberlein et al (MERADCOM), "Improved Fuel Air Explosives," (U) U.S. Army Mobility and Equipment R&D Command Report 2222, Sep. 1977 (S).
3. Frederick H. Gregory, "Finite Element Modeling of the Vulnerability of an M-15 Land Mine Using an Explicit Integration Scheme," Proceedings of the 1981 Army Numerical Analysis and Computers Conference, ARO Report 81-3, Aug. 1981.
4. "ADINA, A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis," ADINA Engineering, Inc., Watertown, MA, Report AE81-1, Sep. 1981.
5. K. J. Bathe, "Static and Dynamic Geometric and Material Nonlinear Analysis Using ADINA," MIT-82448-2, May 1977.
6. M. S. Chawla and R. B. Frey, "A Numerical Study of Projectile Impact on Explosives," BRLMR-2741, Apr. 1977.
7. C. A. Hogentogler, Engineering Properties of Soil, First Edition, McGraw Hill Book Co., 1937, p. 223.
8. John E. Crawford, Private Communication, Aerospace Corporation, El Segundo, CA, March 1982.
9. A. D. Gupta, J. M. Santiago, and H. L. Wisniewski, "An Improved Strain Hardening Characterization in the ADINA Code Using the Mechanical Sublayer Concept," Proceedings of the 1st Chautauqua on Finite Element Modeling, Harwichport, MA, Sep 15-17, 1980.

NUMERICAL RESULTS OF TRANSIENT TWO-DIMENSIONAL HEAT CONDUCTION

R. Yalamanchili
Armament Division
Fire Control & Small Caliber Weapon Systems Laboratory
US Army Armament Research & Development Command
Dover, NJ 07801

ABSTRACT. A general expression is chosen, based on a prior research, for a transient two-dimensional heat conduction. The objective is to choose a variety for Laplacien term, both implicit and explicit finite - differences, and finite - element results. This is programmed in Fortran IV for a digital computer solution. The transient temperatures are predicted due to a step change in boundary conditions for a two-dimensional plate. The midpoint temperatures are discussed in detail where the influence of boundary conditions are minimum. The results are predicted for four Laplacian (different) approximations, explicit, Crank - Nicholson and standard implicit finite - differences and finite - element approaches.

1. INTRODUCTION. Even though the exact analytical solution of heat conduction problems is desirable because these solutions are not only more accurate, but also more explicit in parametric representations, it is frequently necessary to settle for numerical solutions due to complex geometry and / or nonlinear material properties or boundary conditions. An excellent collection of exact solutions for linear problems with simple geometry, such as rectangles, cylinders and spheres, are available in the literature (1). If the exact solution is complex, such as an infinite series solution, the computer programming is still required and at the same time valid only for a specific problem and also may not show explicitly the effects of parameters. If the geometry is complex, such as rifling and variable wall thickness in a gun barrel or multi-layer variable property solid, there is no other resort but application of numerical methods.

Various numerical methods have been used for solutions to the problems of transient heat conduction. The most common are the finite difference method (which represents a direct approximation of the governing partial differential equation) and the finite element method introduced by Wilson and Nickell (10) based on a variational principle derived by Gurtin(3). The finite element method is completely general with respect to geometry and material properties. Complex bodies composed of many different anisotropic materials are easily represented in this method. Temperatures or heat flux boundary conditions may be specified at any point within the finite element system. Moreover, mathematically the method can be shown to converge toward the exact solution as the number of elements is increased. In spite of all these advantages, this method has not been used extensively to solve transient heatconduction problems with radiation boundary conditions.

There are various versions of finite difference approximations to the transient heat conduction problems. However, all these schemes can be classified as either explicit or implicit type. In the case of explicit scheme, the unknowns are determined one at a time by the use of known quantities at one time-step earlier and/or at nodes (already computed for the present time). If

implicit, one equation for each node is to be generated in the entire region of interest and finally, simultaneous solution of all these equations is required. Therefore, the computational times for implicit schemes may be about three times over that of explicit schemes. Numerous difference formulas can be formulated for first-and second-order derivatives and also for the Laplacian term, dependent upon the number of nodes available for taking a derivative, and also on the weights of the data at those nodes. Usually, the difference formulas are constructed for first-and second-order derivatives by the expansion of the function in Taylor's series and the use of an elimination process. The Laplacian term is evaluated by use of the central difference formulas for the second-order derivatives.

The method of weighted residuals (MWR) unifies many approximate methods of the solution of differential equations that are in use today. For unsteady heat conduction, the finite element method and the usual finite difference method were shown (12,13,14) to be special instances of the MWR with a general weighting function. In a more formal way (5,6), variational principles proposed by several authors are all applications of the MWR. An excellent book in the use of the MWR was published recently by Finlayson(4). In literature, this technique is commonly called the error distribution principle. The choice of approximating functions in an assumed solution form is crucial in applying the MWR. No way presently seems to be available to select the approximating functions systematically for all problems. Selection of approximating functions remains somewhat dependent upon the user's intuition and experience, and this is often regarded as a major disadvantage of MWR. Crandall(2) stated that the variation between results obtained by application of different weighting functions to the same approximate solution is much less significant than the variations that can result from the choice of different approximate solutions. Sometimes, one can obtain the exact solution by use of the MWR, if the right choice is made in the selection of the approximate solution form.

2. LAPLACIAN APPROXIMATIONS. Various finite-difference approximations can be derived for the Laplacian term. Consider the rectangular coordinate systems (x,y) and (x',y') separated by an angle, B as shown in Fig. 1. The relations of various quantities between the two coordinate systems (x,y) and (x',y') can be expressed as shown below:

$$x = x' \cos B - y' \sin B, \quad y = x' \sin B + y' \cos B$$

$$\frac{\partial T}{\partial x'} = \cos B \frac{\partial T}{\partial x} + \sin B \frac{\partial T}{\partial y}, \quad \frac{\partial^2 T}{\partial x'^2} = \cos^2 B \frac{\partial^2 T}{\partial x^2} + 2 \cos B \sin B \frac{\partial^2 T}{\partial x \partial y} + \sin^2 B \frac{\partial^2 T}{\partial y^2}$$

$$\frac{\partial T}{\partial y'} = -\sin B \frac{\partial T}{\partial x} + \cos B \frac{\partial T}{\partial y}, \quad \frac{\partial^2 T}{\partial y'^2} = \sin^2 B \frac{\partial^2 T}{\partial x^2} - 2 \cos B \sin B \frac{\partial^2 T}{\partial x \partial y} + \cos^2 B \frac{\partial^2 T}{\partial y^2}$$

$$\frac{\partial^2 T}{\partial a^2} = \frac{\partial^2 T}{\partial x^2}, \quad \frac{\partial^2 T}{\partial b^2} = (1/2) \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial x \partial y} + (1/2) \frac{\partial^2 T}{\partial y^2}$$

$$\frac{\partial^2 T}{\partial c^2} = \frac{\partial^2 T}{\partial y^2}, \quad \frac{\partial^2 T}{\partial d^2} = (1/2) \frac{\partial^2 T}{\partial x^2} - \frac{\partial^2 T}{\partial x \partial y} + (1/2) \frac{\partial^2 T}{\partial y^2}$$

One can derive a second-order derivative by Taylor's series expansion and central differences. For example,

$$\frac{\partial^2 T}{\partial x^2} = (T_{i+1,j} - 2T_{i,j} + T_{i-1,j})/\Delta x^2$$

The subscripts 'i' and 'j' denote respectively the x and y directions or coordinates $\Delta x = \Delta y = h$, for convenience. Now, one can derive five-point finite-difference approximation for Laplacian term as shown below:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{\partial^2 T}{\partial a^2} + \frac{\partial^2 T}{\partial c^2} = L16 = (T_{i-1,j} + T_{i,j-1} + T_{i,j+1} + T_{i+1,j} - 4T_{i,j})/h^2$$

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{\partial^2 T}{\partial b^2} + \frac{\partial^2 T}{\partial d^2} = L17 = (T_{i-1,j-1} + T_{i-1,j+1} + T_{i+1,j-1} + T_{i+1,j+1} - 4T_{i,j})/(2h^2)$$

Similar nine-point approximations are as follows:

$$\begin{aligned} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} &= (1/3)(2\frac{\partial^2 T}{\partial a^2} + \frac{\partial^2 T}{\partial b^2} + 2\frac{\partial^2 T}{\partial c^2} + \frac{\partial^2 T}{\partial d^2}) = L19 \\ &= (T_{i-1,j-1} + 4T_{i-1,j} + T_{i-1,j+1} + 4T_{i,j-1} - 20T_{i,j} + 4T_{i,j+1} + T_{i+1,j-1} \\ &\quad + 4T_{i+1,j} + T_{i+1,j+1})/6h^2 \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} &= (1/3)(\frac{\partial^2 T}{\partial a^2} + 2\frac{\partial^2 T}{\partial b^2} + \frac{\partial^2 T}{\partial c^2} + 2\frac{\partial^2 T}{\partial d^2}) = L17 \text{ 19} \\ &= (T_{i-1,j-1} + T_{i-1,j} + T_{i-1,j+1} + T_{i,j-1} - 8T_{i,j} + T_{i,j+1} + T_{i+1,j-1} \\ &\quad + T_{i+1,j} + T_{i+1,j+1})/3h^2 \end{aligned}$$

L17 19 is not listed, to the author's best knowledge, in any literature. By the way, one can also obtain L17 19 by consideration of arithmetic average of L17 and L19. Numerous approximations can be derived not only by this approach but also by other methods.

3. UNIFICATION. Lemmon and Heaton(7) showed that for unsteady one-dimensional heat conduction problems, the finite-element method and the finite-difference method are the special cases of the MWR. Yalamanchili and Chu (12) derived difference equations for transient two-dimensional heat conduction problems by all three techniques (i.e., finite-element, finite-difference, and MWR), and they were able to bring them into the same format. The above Laplacian term approximations were considered in addition to another one which is not consistent. The subscript 'K' is not used on these equations to indicate time. Appropriate additional subscripts should be introduced into these approximations before final difference equations can be obtained. Even though the application of MWR-Collocation yields (12) finite-difference equations by the use of any Laplacian term approximations, this is not the case for finite-element difference equation. The weighting function is the Dirac-Delta function for MWR-Collocation. The approximate solution form chosen for the method of weighted residuals (MWR) is in the form of sum of products of nodal temperatures and spatial distribution functions. The nodal temperatures are the unknowns; therefore, the weighting function will be the spatial distribution function for the MWR-Galerkin Method. If only L17 19 is chosen as the Laplacian term approximation, one can easily prove (11) that the MWR-Collocation yields finite-difference equation and the MWR-Galerkin yields finite-element difference equation. Therefore, one can conclude that the finite-element and finite-difference methods belong to the class of MWR.

4. ACCURACY. The comparison of accuracy of various Laplacian term approximations mentioned above, in terms of order of magnitude, reveal: L16 is the most accurate 5-point formula; L17 19 uses nine points with no increase in accuracy; L17 is worse than L16; and L19 is by far the most accurate. It is important to remember that the number of computations either in generating the coefficient matrix of the system of algebraic equations or in their solution procedures increases significantly with the increase in the number of points in the differ-

ence equation. Since L16 and L17 19 are of the same accuracy, naturally L16 may be preferred because of the less number of points involved. Therefore, the finite element method can not be the best as far as accuracy is concerned because L19 is by far the most accurate from theoretical considerations.

5. NUMERICAL EXPERIMENTS. The difference equation may be summarized into the following form whether it is based on finite element, finite difference, MWR-Collocation, or MWR-Galerkin methods:

$$\begin{aligned}
 & AT_{i-1,j-1,k+1} + BT_{i-1,j,k+1} + AT_{i-1,j+1,k+1} \\
 & + BT_{i,j-1,k+1} + CT_{i,j,k+1} + BT_{i,j+1,k+1} + AT_{i+1,j-1,k+1} \\
 & + BT_{i+1,j,k+1} + AT_{i+1,j+1,k+1} = DT_{i-1,j-1,k} + ET_{i-1,j,k} \\
 & + DT_{i-1,j+1,k} + ET_{i,j-1,k} + FT_{i,j,k} + ET_{i,j+1,k} \\
 & + DT_{i+1,j-1,k} + ET_{i+1,j,k} + DT_{i+1,j+1,k}
 \end{aligned}$$

However, the coefficients A,B,C,D,E, and F are different depending upon the Laplacian term approximation and the MWR-Collocation or MWR-Galerkin. These are shown in Table 1 and also in stencil form in Figure 1. The parameter in this equation allows a weighted average of the sum of two second order spatial derivatives at two discrete times. An explicit scheme can result when ϕ is set to zero; otherwise, an implicit scheme will be the result for the remaining range of parameter, ϕ .

TABLE 1 - COEFFICIENTS OF VARIOUS DIFFERENCE EXPRESSIONS

METHOD	COEFFICIENTS					
	A	B	C	D	E	F
F E	$\frac{1}{36} - \frac{1}{3} \phi$	$\frac{1}{9} - \frac{1}{3} \phi$	$\frac{4}{9} + \frac{8}{3} \phi$	$\frac{1}{36} + \frac{1}{3} \phi$	$\frac{1}{9} + \frac{1}{3} \phi$	$\frac{4}{9} - \frac{8}{3} \phi$
FD/MWR-C						
L 17 19	$-\frac{2}{3} \phi$	$-\frac{2}{3} \phi$	$1 + \frac{16}{3} \phi$	$\frac{2}{3} (1 - \phi)$	$\frac{2}{3} (1 - \phi)$	$1 - \frac{16}{3} (1 - \phi)$
L 16	0	-2ϕ	$1 + 8 \phi$	0	$2 (1 - \phi)$	$1 - 8 (1 - \phi)$
L 17	$-\phi$	0	$1 + 4 \phi$	$(1 - \phi)$	0	$1 - 4 (1 - \phi)$
L 19	$-\frac{4}{3} \phi$	$-\frac{4}{3} \phi$	$1 + \frac{20}{3} \phi$	$(1 - \phi) \frac{8}{3}$	$\frac{4}{3} (1 - \phi)$	$1 - \frac{20}{3} (1 - \phi)$
MWR-G						
L 17 19	$\frac{1}{36} - \frac{2}{3} \phi$	$\frac{1}{9} - \frac{2}{3} \phi$	$\frac{4}{9} + \frac{16}{3} \phi$	$\frac{1}{36} + \frac{2}{3} (1 - \phi)$	$\frac{1}{9} + \frac{2}{3} (1 - \phi)$	$\frac{4}{9} - \frac{16}{3} (1 - \phi)$
L 16	$\frac{1}{36}$	$\frac{1}{9} - 2 \phi$	$\frac{4}{9} + 8 \phi$	$\frac{1}{36}$	$\frac{1}{9} + 2 (1 - \phi)$	$\frac{4}{9} - 8 (1 - \phi)$
L 17	$\frac{1}{36} - \phi$	$\frac{1}{9}$	$\frac{4}{9} + 4 \phi$	$\frac{1}{36} + (1 - \phi)$	$\frac{1}{9}$	$\frac{4}{9} - 4 (1 - \phi)$
L 19	$\frac{1}{36} - \frac{1}{3} \phi$	$\frac{1}{9} - \frac{4}{3} \phi$	$\frac{4}{9} + \frac{20}{3} \phi$	$\frac{1}{36} + \frac{1}{3} (1 - \phi)$	$\frac{1}{9} + \frac{4}{3} (1 - \phi)$	$\frac{4}{9} - \frac{20}{3} (1 - \phi)$

TWO-DIMENSIONAL ELEMENTS AROUND (x_i, y_i)

TABLE 2 - STABILITY AND OSCILLATION LIMITS

297

These are also available elsewhere (11) for arbitrary values of parameter, ϕ . In general, the MWR-Galerkin requires less step sizes than the MWR-Collocation or finite differences to enforce stability or nonoscillations or both. Also, the Dusenberre criteria is more conservative than the general or VonNeumann stability criteria. The nonoscillatory criteria is more restrictive than any stability criteria. Even though the finite element method is unconditionally stable according to general or VonNeumann criteria, there is a limit due to Dusenberre criteria and also another further lower limit for nonoscillations.

It is possible to generate either completely explicit scheme ($\phi=0$) or standard implicit scheme ($\phi=1$) or numerous other implicit schemes for arbitrary values of parameter ϕ ($0 < \phi < 1$). The coefficient matrix of these systems of equations is symmetric and also banded. Special numbering system for the identification of unknown nodal temperatures minimized the width of the banded matrix. The storage required for nonzero elements and also the number of computations will be minimum if the band width is minimum. The typical matrix is of the type, 100 by 100. However, this is a sparse matrix and the nonzero elements are in a block-tridiagonal matrix form. Therefore, the components of unknown vector can be grouped into subsets, and these subsets can be eliminated, as in the Gaussian procedure, a group at a time. Here, the coefficient matrix is decomposed into upper triangular matrix and a lower triangular matrix, each one will be in block-bidiagonal matrix form. There is also another procedure for sparse matrices where a specified number of lower and upper diagonals have nonzero elements. Solution procedure is done by means of Gauss elimination with column pivoting. Optimization of these algorithms, for core storage and number of computations, is essential because of repetition for each time-step.

6. EXAMPLE. A two-dimensional plate, 2 by 2, with a unit thermal diffusivity is considered. Initially, the temperature is zero everywhere. Suddenly, all four sides are exposed to unit temperature. The temperature distribution is determined as a function of time until steady state (0.8) is reached. The midpoint temperatures are shown in Table 3. The data marked asterisk (*) is more accurate than the other data at that time. This notation is used in all the tables.

None of the schemes maintained the same accuracy for all times. Therefore, one can never tell which one will be more accurate at any given time. Even though there is tremendous difference in truncation error between $L19 \left(\frac{h^6}{504} \frac{\partial^6 T}{\partial x^4 \partial y^4} + o(h^8) \right)$ and $L17 \ 19 \left(\frac{h^2}{96} \frac{\partial^4 T}{\partial x^2 \partial y^2} + o(h^4) \right)$, the experiments show very little difference in results between the two approximations either in MWR-Collocation or MWR-Galerkin. For small times (such as 0.05), the MWR-Galerkin predicts the results negative, very low and off at least by an order of magnitude. Therefore, the MWR-Collocation or finite differences should be preferred for small times. Since the MWR-Galerkin is not accurate for small times, the finite element method ($\phi=\frac{1}{2}$, L1719, MWR-Galerkin) is not desirable for small times.

TABLE 3 - MDIPOINT TEMPERATURES FOR $\Delta L = .2$, $\Delta t/\Delta L^2 = .25$, $\phi = .5$

Method	Laplacian	Midpoint Temperature at Time				
		.05	.1	.2	.4	.75
MWR-C/FD	L16	.01601	.1159	.4106	.7751*	.9594*
	L17	.01539	.1111	.3978	.7639	.9556
	L19	.01589*	.1145	.4064	.7715	.9582
	L1719	.01570	.1130	.4023*	.7678	.9569
MWR-G	L16	-.00160	.1023	.4245	.7893	.9641
	L17	-.00178	.09731	.4111	.7786	.9606
	L19	-.00150	.1007	.4200	.7858	.9630
MWR-G/FE	L1719	-.00152	.09906*	.4155	.7822	.9618
ADI (9)		.01579	.09333	.40354	.77532	.96003
Exact		-	.09883	.40354	.77486	-

The midpoint temperatures for large time-steps are shown in Table 4. Most of the conclusions mentioned above remained the same. The MWR-Galerkin under estimates for one-quarter of steady state times and over estimates for the remaining transient period.

TABLE 4 - MIDPOINT TEMPERATURES FOR $\Delta L = .2$, $\Delta t/\Delta L^2 = 2.5$, $\phi = .5$

Method	Laplacian	Midpoint Temperature at Time			
		.1	.2	.4	.7
MWR-C/FD	L16	.0969*	.3773*	.7843	.9505
	L17	.0934	.3653	.7729*	.9461
	L19	.0959	.3735	.7806	.9491
	L1719	.0947	.3696	.7768	.9477
MWR-G	L16	.0839	.3769	.7951	.9498
	L17	.0804	.3646	.7850	.9465
	L19	.0829	.3728	.7919	.9488
MWR-G/FE	L1719	.0818	.3686	.7885	.9477

The midpoint temperatures for standard implicit schemes are shown in Table 5. The step sizes are same for the results in Tables 4 and 5. However, the results for small times, such as 0.1, are very bad for standard implicit scheme ($\phi = 1$). These are higher by about 50 percent. In general, the results are higher for small times and lower for large times. The cross over point may be about one-quarter of steady state time. The optimization of ϕ with respect to step sizes may very well improve the accuracy significantly.

TABLE 5 - MIDPOINT TEMPERATURES FOR $\Delta L = .2$, $\Delta t/\Delta L^2 = 2.5$, $\phi = 1$

Method	Laplacian	Midpoint Temperature at Time			
		.1	.2	.4	.7
MWR-C/FD	L16	.1514	.3572	.6838	.9022
	L17	.1470	.3488	.6739	.8965
	L19	.1500	.3545	.6806	.9003
	L1719	.1486	.3518	.6773	.8984
MWR-G	L16	.1475	.3600*	.6949*	.9093*
	L17	.1430*	.3515	.6851	.9039
	L19	.1461	.3572	.6917	.9075
	L1719	.1446	.3544	.6884	.9057

The results are supposed to be nonoscillatory and also stable irrespective of any criteria, according to theory (11) and Table 2. Even though the step sizes are large for the results of Tables 4 and 5, further larger time-steps are used to demonstrate this hypothesis. The results are shown in Table 6. The trend is similar to Table 5 as far as accuracy is concerned.

TABLE 6 - MIDPOINT TEMPERATURES FOR $\Delta L = .2$, $\Delta t/\Delta L^2 = 5$, $\phi = 1$

Method	Laplacian	Midpoint Temperatures at Time			
		.2	.4	.6	.8
MWR-C/FD	L16	.3279	.6198	.7994	.8970
	L17	.3212	.6112	.7921	.8918
	L19	.3258	.6170	.7970	.8953
	L1719	.3236	.6142	.7947	.8936
MWR-G	L16	.3290*	.6275*	.8077*	.9032*
	L17	.3223	.6189	.8006	.8983
	L19	.3268	.6247	.8053	.9016
	L1719	.3246	.6218	.8030	.9000

The midpoint temperatures for explicit schemes are shown in Table 7. Due to the use of MWR-Galerkin, all the results are meaningless because of unbounded oscillations and thus unstable characteristics. The computations are performed only 5, 13, 6, and 9 time-steps before the absolute midpoint temperature exceeded a certain arbitrary value due to L16, L17, L19, and L1719 respectively. Thus, L17 may be considered as less oscillatory and more stable than the other Laplacian approximations. This conclusion is identical to theoretical predictions. None of the approximations are good at time, .05, due to MWR-Collocation or finite differences. For other times, L17 is the most accurate of all. The results are supposed to be oscillatory according to theory even for MWR-Collocation/Finite-Differences except due to L17 and L1719. But, this is not the case in practice. However, all are stable from both theory and experiment.

TABLE 7 - MIDPOINT TEMPERATURES FOR $\Delta L = .2$, $\Delta t / \Delta L^2 = .25$, $\phi = 0$

Method	Laplacian	Midpoint Temperature at Time				
		.05	.1	.2	.4	.75
MWR-C/FD	L16	.0039	.1055	.4204	.7857	.9630
	L17	.0038	.1010*	.4070*	.7746*	.9593
	L19	.0039	.1043	.4163	.7822	.9618
	L1719	.0039	.1028	.4119	.7785	.9606*

The time-step is doubled to single out further the most nonoscillatory and stable scheme and also accurate. The results are shown in Table 8. The MWR-Collocation or finite differences and L17 may be such a scheme. None of the Laplacian term approximations are good for small time. The results are supposed to be oscillatory for all methods and Laplacian term approximations. However, L17 and L1719 escaped such disturbances. Thus one doesn't have to enforce strictly the limits given in Table 2. L1719 is supposed to be unstable according to Dusiinberre. However, it is not the case as shown by numerical experiments. Therefore, the Dusiinberre criteria may be more conservative than in reality. In the case of MWR-Collocation or finite differences, the number of time steps required to exceed a certain arbitrary value are 13 and 26 respectively for L16 and L19. The similar order is 4,5,5,4 for L16, L17, L19, L1719 and MWR-Galerkin.

TABLE 8 -MIDPOINT TEMPERATURES FOR $\Delta L = .2$, $\Delta t / \Delta L^2 = .5$, $\phi = 0$

Method	Laplacian	Midpoint Temperature at Time				
		.06	.1	.2	.4	.76
MWR-C/FD	L17	0	.1211*	.3896*	.7749*	.9630
	L1719	0	.1245	.3864	.7686	.9594*

7. CONCLUSIONS. Various numerical methods, in particular, finite element (FE), finite difference (FD) and weighted residuals methods (MWR), are reviewed. It was shown that the MWR-Collocation yields FD whereas the MWR-Galerkin yields FE and thus it may be concluded that the FE and FD belong to the class of MWR. Several Laplacian terms, including the most accurate 5-node and 9-node formulas as well as another new one (L1719) that unified FE,FD, and MWR; were examined by order of magnitude analysis to compare their accuracy. The finite element method can not be the best as far as accuracy is concerned. Numerous difference equations are given and the numerical details of generating the system of equations, their characteristics, and their solution procedures are discussed. The stability and nonoscillation limits are given in the form of table for explicit, Crank-Nicholson type ($\phi = 0.5$), and standard implicit schemes. They do point out that the most accurate scheme doesn't necessarily possess the best stability and nonoscillation characteristics. It is also clear that the MWR-Galerkin requires less step sizes than the MWR-Collocation or finite differences to enforce stability or nonoscillations or both.

The transient temperatures are predicted due to a step change in boundary conditions for a two-dimensional plate. The numerical experiments are performed with all five Laplacian term approximations; the MWR-Collocation or finite-differences; the MWR-Galerkin/finite-element; and also explicit, Crank-Nicholson type, and implicit types to observe the accuracy, stability, and oscillation characteristics. The midpoint temperatures are only tabulated to avoid the influence of boundary conditions.

None of the schemes maintained the same accuracy at all times. Therefore, one can never tell which one will be more accurate at any given time. Even though there is tremendous differences in truncation error between L19 and L1719, the experiments show very little difference in results between the two approximations. The MWR-Galerkin/finite-element underestimates for small times and thus these may not be desirable for several time steps. The standard implicit scheme ($\phi=1$) overestimates for small times by about 50 percent and thus Crank-Nicholson type ($\phi = .5$) is preferred for initiation of computations at least for several time steps. The temperatures are lower for large times with standard implicit scheme. The optimization of ϕ with respect to step sizes may very well improve the accuracy significantly. It is confirmed that the results are nonoscillatory and stable for standard implicit schemes. L17 is found to be less oscillatory and more stable and also more accurate than the other Laplacian approximations for explicit schemes. The superiority of L17 in nonoscillatory and stable characteristics is also confirmed by theory. There is a difference, in break-even or maximum step sizes, between theory and practice to enforce stability or nonoscillations or both. Thus, slightly higher than the step sizes given by theory may be utilized in practice. The Dusenberre stability criteria may be more conservative than in reality. In general, the MWR-Collocation or finite-differences is better than the MWR-Galerkin or finite-element.

8. REFERENCES.

1. Carslaw, H.S., and Jaeger, J.C., Conduction of Heat in Solids, Oxford University Press, England, 1959.
2. Crandall, S.H., Engineering Analysis, McGraw Hill, New York, N.Y. 1956.
3. Gurtin, M.E., "Variational Principles for Linear Initial Value Problems," Quarterly Journal of Applied Mathematics, Vol. 22, 1964, pp 252-256.
4. Finlayson, B.A., The Method of Weighted Residuals and Variational Principles With Application in Fluid Mechanics, Heat and Mass Transfer, Academic Press, New York, N.Y. 1972.
5. Finlayson B.A., and Scriven, L.E., "The Method of Weighted Residuals-A Review," Applied Mechanics Reviews, Vol. 19, No. 9, 1966, pp. 735-748.
6. Finlayson, B.A., and Scriven, L.E., "The Method of Weighted Residuals and Its Relation to Certain Variational Principles for the Analysis of Transport Processes," Chemical Engineering Science, Vol. 20, 1965, pp. 395-404.
7. Lemmon, E.C., and Heaton, H.S., "Accuracy, Stability, and Oscillation Characteristics of Finite Element Method for Solving Heat Conduction Equations," ASME Paper #69-WA/HT-35.
8. Milne, W.E., Numerical Solution of Differential Equations, 1st Edition, John Wiley & Sons, New York, 1953; 2nd Edition, Dover Publications, New York 1970.
9. Peacemen, S.W., and Rachford, H.H., JR., "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. of SIAM, Vol. 3, 1955, pp. 28-41.
10. Wilson, W.L., and Nickell, R.E., "Application of the Finite-Element Method to Heat Conduction Analysis," Nuclear Engineering and Design, Vol. 4, 1966, pp. 276-286.

11. Yalamanchili, R., "Accuracy, Stability, and Oscillation Characteristics of Transient Two-Dimensional Heat Conduction," ASME Paper #75-WA/HT-85, Dec, 1975.
12. Yalamanchili, R., and Chu, S.C., "Stability and Oscillation Characteristics of Finite-Element, Finite-Difference, and Weighted-Residuals Methods for Transient Two-Dimensional Heat Conduction in Solids," J. of Heat Transfer, Vol. 95 Series c, No. 2, May 1973.
13. Yalamanchili, R., and Chu, S.C., "Application of the Finite Element Method To Heat Transfer Problems, Part II-Transient Two-Dimensional Heat Transfer With Convection and Radiation Boundary Conditions," Technical Report RE-TR-71-41, Research Directorate, U.S. Army Weapons Command, Rock Island, IL (AD726371), 1971.
14. Yalamanchili, R., and Chu, S.C., "Finite Element Method to Transient Two-Dimensional Heat Transfer With Convection and Radiation Boundary Conditions, U.S. Army Weapons Command, Technical Report RE-70-165 (AD 709604), 1970.

SALOME, A STRUCTURED AND LOGICALLY MINIMAL ENSEMBLE
OF PROGRAMMING CONSTRUCTS

Royce W. Soanes, Jr.
U.S. Army Armament Research and Development Command
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189

ABSTRACT. An overview of the Salome structured programming language will be given, along with an appendix briefly describing its syntax and semantics. Salome has been designed with the Fortran programmer in mind. The Salome source language is supported by a translator having a target language of standard Fortran, and Fortran code may be injected into Salome source code when needed. The Salome translator has been written in Salome and Fortran.

1. OVERVIEW. The acronym SALOME stands for structured and logically minimal ensemble (or selection and looping operations made easy). The term "structured" refers to the gotoless nature of the logical control statements of Salome. While the power of the goto statement has corrupted many programs, the logical control statements of Salome virtually eliminate the need to use goto statements and labels in programs. The phrase "logically minimal" refers to the fact that the variety and semantic ambiguity of the structured logical control statements is kept to a minimum.

There is one looping construct and one selection construct in Salome, while other languages may force the programmer to learn several. Flexibility and unambiguous semantics are stressed in Salome - not variety.

Blanks are as important in Salome as they are in ordinary English text. The use of blanks as delimiters enhances readability, eliminates extraneous punctuation, and allows for greater brevity of syntax. Although Salome is delimiter oriented (as opposed to line oriented) there is no general end of statement or between statement delimiter in Salome. A small penalty one pays for this feature is that blanks are not allowed in assignment statements. This was considered to be a small price to pay for the elimination of a lot of extraneous semicolons.

The Salome language is supported by a one pass translator whose target language is Fortran, hence, when one writes a program in Salome, one is in effect writing in two different high level languages at the same time. The translator is in turn supported by a package of Fortran callable string manipulation routines. The primary reason Fortran was selected as the target language was to benefit those unfortunate programmers who are inextricably mired down in a Fortran environment when Fortran is more lacking in structured programming constructs than any other high level language except Basic. A widely used high level target language also assures greater portability of Salome and its translator.

The file produced by the Salome translator consists of the original Salome source code inserted as special Fortran comments interleaved with the generated Fortran code. When Salome syntax errors are caught, they are pointed out by a string of dots. If there are syntax errors in a Salome routine, only the first one in that routine is flagged and the rest of the code in that routine is ignored. No attempt at error recovery is made because one can't be absolutely positive about what the programmer actually intended, and guessing usually uncovers syntax errors that aren't.

The reason for the interleaving of the Salome source code as comments with the generated Fortran code is to enable the programmer to relate any syntax errors picked up by the Fortran compiler (and not picked up by the Salome translator) back to the original Salome source code. The Salome translator does not check for things that the Fortran compiler is going to check for anyway.

Since the appendix of this paper contains a brief but fairly complete description of Salome, no further details will be described here. Instead, a small subroutine written in Salome along with the generated Fortran and the interleaved file are presented as a small exercise in deduction. If one knows Fortran and one reads the interleaved file carefully, one can deduce the exact semantic meaning of the Salome routine.

SALOME SOURCE

```
SUB LININT ( N X Y XI YI ) -- -----LININT
-- LINEAR INTERPOLATION
(
--
N=NO. OF POINTS.
X=ABSCISSA ARRAY.
Y=ORDINATE ARRAY.
XI=ABSCISSA AT WHICH INTERPOLATION IS DESIRED.
YI=INTERPOLATED RESULT.
X IS ASSUMED TO BE SORTED IN ASCENDING ORDER.
--)
DIM X 1 , Y 1 .
-- ASSUME AT LEAST 2 POINTS OF DATA
@ N > 1 @
-- GET PROPER SUBINTERVAL USING BINARY SEARCH
IL=1 IR=N
DO # IL+1 = IR #
-- COMPUTE INDEX OF ABSCISSA MIDWAY BETWEEN IL AND IR
IM=(IL+IR)/2
-- REDEFINE IL OR IR AS IM
IF XI < X(IM) , IR=IM ; IL=IM FI
-- ASSUME IL AND IR ARE STILL IN PROPER ORDER
@ IL < IR @ OD
DX=X(IR)-X(IL) DY=Y(IR)-Y(IL)
-- ASSUME LENGTH OF SUBINTERVAL IS POSITIVE
@ DX > 0. @
YI=Y(IL)+(DY/DX)*(XI-X(IL))
RET END
```

GENERATED FORTRAN

```

SUBROUTINE LININT(N,X,Y,XI,YI)
C -----LININT
C   LINEAR INTERPOLATION
C
C   N=NO. OF POINTS.
C   X=ABSCISSA ARRAY.
C   Y=ORDINATE ARRAY.
C   XI=ABSCISSA AT WHICH INTERPOLATION IS DESIRED.
C   YI=INTERPOLATED RESULT.
C   X IS ASSUMED TO BE SORTED IN ASCENDING ORDER.
C
C   DIMENSION X(1),Y(1)
C   ASSUME AT LEAST 2 POINTS OF DATA
C   IF(N.GT.1) GO TO 1000
C   WRITE(6,1001)
1001 FORMAT('/////,' N > 1 IS FALSE IN LININT')
C   CALL EXIT
C   GET PROPER SUBINTERVAL USING BINARY SEARCH
1000 IL=1
C   IR=N
1002 IF(IL+1.EQ.IR) GO TO 1003
C   COMPUTE INDEX OF ABSCISSA MIDWAY BETWEEN IL AND IR
C   IM=(IL+IR)/2
C   REDEFINE IL OR IR AS IM
C   IF(.NOT.(XI.LT.X(IM))) GO TO 1005
C   IR=IM
C   GO TO 1004
1005 IL=IM
C   ASSUME IL AND IR ARE STILL IN PROPER ORDER
1004 IF(IL.LT.IR) GO TO 1006
C   WRITE(6,1007)
1007 FORMAT('/////,' IL < IR IS FALSE IN LININT')
C   CALL EXIT
1006 GO TO 1002
1003 DX=X(IR)-X(IL)
C   DY=Y(IR)-Y(IL)
C   ASSUME LENGTH OF SUBINTERVAL IS POSITIVE
C   IF(DX.GT.0) GO TO 1008
C   WRITE(6,1009)
1009 FORMAT('/////,' DX > 0. IS FALSE IN LININT')
C   CALL EXIT
1008 YI=Y(IL)+(DY/DX)*(XI-X(IL))
C   RETURN
C   END

```

INTERLEAVED SALOME AND FORTRAN

```

C__S__SUB LININT ( N X Y XI YI ) -----LININT
      SUBROUTINE LININT (N,X,Y,XI,YI)
C      -----LININT
C__S__-- LINEAR INTERPOLATION
C      LINEAR INTERPOLATION
C__S__(--
C
C__S__N=NO. OF POINTS.
C      N=NO. OF POINTS.
C__S__X=ABSCISSA ARRAY.
C      X=ABSCISSA ARRAY.
C__S__Y=ORDINATE ARRAY.
C      Y=ORDINATE ARRAY.
C__S__XI=ABSCISSA AT WHICH INTERPOLATION IS DESIRED.
C      XI=ABSCISSA AT WHICH INTERPOLATION IS DESIRED.
C__S__YI=INTERPOLATED RESULT.
C      YI=INTERPOLATED RESULT.
C__S__X IS ASSUMED TO BE SORTED IN ASCENDING ORDER.
C      X IS ASSUMED TO BE SORTED IN ASCENDING ORDER.
C__S__-- )
C
C__S__DIM X 1 , Y 1 .
      DIMENSION X(1),Y(1)
C__S__-- ASSUME AT LEAST 2 POINTS OF DATA
C      ASSUME AT LEAST 2 POINTS OF DATA
C__S__@ N > 1 @
      IF(N.GT.1) GO TO 1000
      WRITE(6,1001)
1001  FORMAT(////, ' N > 1 IS FALSE IN LININT' )
      CALL EXIT
C__S__-- GET PROPER SUBINTERVAL USING BINARY SEARCH
C      GET PROPER SUBINTERVAL USING BINARY SEARCH
C__S__IL=1 IR=N
1000  IL=1
      IR=N
C__S__DO # IL+1 = IR #
1002  IF(IL+1.EQ.IR) GO TO 1003
C__S__-- COMPUTE INDEX OF ABSCISSA MIDWAY BETWEEN IL AND IR
C      COMPUTE INDEX OF ABSCISSA MIDWAY BETWEEN IL AND IR
C__S__IM=(IL+IR)/2
      IM=(IL+IR)/2
C__S__-- REDEFINE IL OR IR AS IM
C      REDEFINE IL OR IR AS IM
C__S__IF XI < X(IM) , IR=IM ; IL=IM FI
      IF(.NOT.(XI.LT.X(IM))) GO TO 1005
      IR=IM
      GO TO 1004
1005  IL=IM
C__S__-- ASSUME IL AND IR ARE STILL IN PROPER ORDER
C      ASSUME IL AND IR ARE STILL IN PROPER ORDER

```

```

C__S__  @ IL < IR @ OD
1004 IF(IL.LT.IR) GO TO 1006
      WRITE(6,1007)
1007 FORMAT(//////,' IL < IR IS FALSE IN LININT')
      CALL EXIT
1006 GO TO 1002
C__S__DX=X(IR)-X(IL) DY=Y(IR)-Y(IL)
1003 DX=X(IR)-X(IL)
      DY=Y(IR)-Y(IL)
C__S__-- ASSUME LENGTH OF SUBINTERVAL IS POSITIVE
C      ASSUME LENGTH OF SUBINTERVAL IS POSITIVE
C__S__@ DX > 0. @
      IF(DX.GT.0) GO TO 1008
      WRITE(6,1009)
1009 FORMAT(//////,' DX > 0. IS FALSE IN LININT')
      CALL EXIT
C__S__YI=Y(IL)+(DY/DX)*(XI-X(IL))
1008 YI=Y(IL)+(DY/DX)*(XI-X(IL))
C__S__RET END
      RETURN
      END

```

APPENDIX

A QUICK GUIDE TO THE SALOME PROGRAMMING LANGUAGE

1. Definition of Isolated String

An isolated string resides on a single line, is preceded by one or more blanks (or begins in Column 1) and is followed by one or more blanks (or ends in Column 72).

All Salome keywords must be isolated strings containing no blanks.

2. Comments

- A. A 'tack on' comment may stand alone on a line or it may be tacked onto the end of another statement.

Ex.

```
-- TACK ON COMMENT STANDING ALONE  
A=B+C -- TACKED ON TACK ON COMMENT
```

'--' is the keyword for tack on comments

The tack on comment has essentially the same form as the ADA comment.

- B. A delimited comment may occupy more than one line.

Ex.

```
(-- THIS IS  
A DELIMITED  
COMMENT--)
```

'(--' AND '--)' are the keywords for delimited comments.

Delimited comments may be used to temporarily disable certain sections of executable code and they may be nested.

3. Injected Fortran

Fortran statements may be injected into a Salome program when needed.

Ex.

```
F      COMMON A(100),B(50)  
F      A = B * C
```

'F' is the keyword for injecting Fortran. The Fortran statement begins immediately after the blank following 'F'.

4. Variable Declaration Statements

The correspondence between Salome variable declaration keywords and Fortran variable declaration keywords is given by the following table.

SALOME	FORTRAN
DIM	DIMENSION
INT	INTEGER
REAL	REAL
DP	DOUBLE PRECISION
LOG	LOGICAL
EQV	EQUIVALENCE

Ex.

DIM X 10 20 , Y 100 . translates to
DIMENSION X(10,20),Y(100)

INT A , B , C 10 20 30 . translates to
INTEGER A,B,C(10,20,30)

The aforementioned keywords plus ',' and '.' are the keywords for declaration statements.

'.' Ends all variable declaration statements.

',' Separates variable references in all variable declaration statements except the equivalence statement.

',' Separates variable group references in the equivalence statement.

Ex.

EQV A B C , K(5) L M(10) . Translates to
EQUIVALENCE (A,B,C),(K(5),L,M(10))

Note that a string such as A(10,20) may appear in an equivalence statement but not in any other variable declaration statement.

5. Assignment Statements

Salome assignment statements must be isolated strings containing no blanks. They are otherwise equivalent to Fortran assignment statements. If blanks are desired in assignment statements, they may be injected as Fortran.

Ex.

A=B*C+D P=Q*R

F A = B + C * D / E

6. I/O

A. The keywords which begin I/O statements are:

'IN' 'OUT' 'R' 'W' 'RF' 'WF' 'RU' 'WU' 'FMT'

They may be read as follows:

IN - Input on device no.
OUT - Output on device no.
R - Read
W - Write
RF - Read in Format
WF - Write in Format
RU - Read Unformatted
WU - Write Unformatted
FMT - Format

B. The rest of the I/O keywords are:

(' ') 'EOF:' 'ERR:'

(' ' and ') bracket format strings in 'R' and 'W' statements.

'.' ends all I/O statements except for the IN and OUT statements.

'EOF:' precedes any end of file label name.

'ERR:' precedes any error label name.

Ex.

IN 1

OUT IOUT

R I X Y (I3 2F10.0) EOF: END-OF-DATA .

W I X Y (' I=' I3 ' X=' E14.7 ' Y=' E14.7) .

RF F1 A B EOF: ENDDATA .

FMT F1 8F10.0 .

W (' PLAIN HOLLERITH ') .

Note that format strings in 'FMT' statements are not enclosed in parentheses as they are in 'R' and 'W' statements.

Format references in RF, WF, and FMT statements may be any isolated string containing no blanks.

7. External Subroutine Calls

External subroutines are called by writing the name of the subroutine followed by its argument list, if any. No 'call' keyword is used and English Salome keywords may not be used as subroutine names.

Ex.

EXIT - CALL TO EXIT

ADD (A B C) -- ADD A TO B GIVING C

DEFSTR (ABC ' A B C ') -- DEFINE STRING

The only keywords associated with an external subroutine call are '(' and ')'.

Arguments in an external subroutine call must be isolated strings. The only argument which may contain blanks is quoted Hollerith. Arguments are not separated by commas.

8. External Subprogram Declaration Statements

The keywords beginning external subprogram declarations and their Fortran counterparts are given by the following table.

SALOME	FORTRAN
SUB	SUBROUTINE
FUN	FUNCTION
IFUN	INTEGER FUNCTION
RFUN	REAL FUNCTION
LFUN	LOGICAL FUNCTION
DPFUN	DOUBLE PRECISION FUNCTION

'(' and ')' are the only other external subprogram declaration keywords.

Additional keywords necessary to complete the definition of an external subprogram are:

SALOME	FORTRAN
-->	ENTRY
RET	RETURN
END	END

Ex.

SUB INIT -- INITIALIZATION ROUTINE

SUB MULT (A B C) -- MULTIPLY B BY A GIVING C

FUN PROD (X Y) -- Product of X and Y

Parameters in an external subprogram declaration must be isolated strings containing no blanks.

9. Internal Subroutine Calls

External subroutine calls in Salome are quite similar to those of Fortran, but Salome also has the facility for defining and calling internal subroutines. These sections of code are called internal because they belong to the program or subprogram which uses them and they may not be called by any other external subprogram.

An internal subroutine name may be any isolated string.

An internal subroutine is called by enclosing its name in double quotes (not two single quotes). The double quotes are the only keywords associated with the internal subroutine call.

Ex.

```
" READ INPUT DATA "
```

```
" INITIALIZE ARRAYS "
```

No arguments are passed to internal subroutines. All data in the calling program is available to the internal subroutine.

10. Internal Subroutine Declaration Statements

Internal subroutines are declared at the end of the calling program (after a return or call to exit) by writing 'TO' followed by the internal subroutine name enclosed in double quotes, followed by any code, followed by 'OT'.

Ex.

```
EXIT ( OR RET )  
TO " READ INPUT DATA "  
  R X Y Z ( 3F10.0 ) . OT
```

Internal subroutines may call other internal subroutines within the same program or subprogram.

Internal subroutines have exactly one exit point at 'OT'.

The Salome internal subroutine simplifies the simulation of recursion considerably.

11. If Statement

The 6 keywords involved in the If Statement are:

'IF' ',' '\$' '/' ';' 'FI'

The basic functions of these keywords during translation time and run time are:

'IF' Tells the translator that the first Boolean expression is about to begin.

',' Ends a Boolean expression and selects the statement sequence following to be executed if the Boolean is true. This keyword may be read: 'then'.

'\$' Ends a Boolean expression and selects the statement sequence following to be executed if the Boolean is false. This keyword may be read: 'Then Don't Select'.

'/' Halts execution in the if statement if the previous statement sequence has been executed and tells the translator that another Boolean expression is about to begin. This keyword may be read: 'Otherwise,If'.

';' Halts execution in the if statement if the previous statement sequence has been executed and selects the statement sequence following to be executed if no other statement sequence has yet been executed. This keyword may be read 'Otherwise'.

'FI' ends the If Statement.

The Relational and Boolean operators used in Boolean expressions are:

'<' '>' '<=' '>=' '=' '/=' 'AND' 'OR' 'NOT'

Relational operators must not contain blanks, but they need not be isolated. The Boolean operators 'AND', 'OR' and 'NOT' must be isolated and not contain blanks.

The general form of the If Statement is as follows:

```
IF B1 , S1 /  
   B2 , S2 /  
   B3 , S3 /  
   .  
   .  
   B1 , S1 /  
   .  
   .  
   BN , SN ; S FI
```

Where B1 through BN are Boolean expressions and S1 through SN and S are sequences of statements.

If Statements may be nested to theoretically any level.

Ex.

```
IF A>0. AND B>0. , X=A*B ; X=0. FI
```

```
IF I/=0 , X=0. Y=0. Z=0. FI
```

```
IF X < A , F=-1. /  
  X > B , F=1. ; F=0. FI
```

```
IF I<1 , I=1 /  
  I>N , I=N FI
```

12. Assertion or Assumption Statements

One may very quickly insert error checking code into a program via the assumption or assertion statement.

This statement consists of nothing more than a Boolean expression delimited on both sides by the keyword '@'.

Ex.

```
SUB INTERP ( N X Y XI YI ) — INTERPOLATION ROUTINE  
.  
.  
@ N>1 @  
.  
.  
END
```

The assumption that $N > 1$ in subroutine interp will translate into the following typical Fortran code.

```
IF(N.GT.1) GO TO 1000  
WRITE(6,1001)  
1001 FORMAT(////,' N>1 IS FALSE IN INTERP')  
CALL EXIT  
1000 CONTINUE
```

The Salome assertion statement is considerably less verbose than the corresponding Fortran error checking code and will therefore make it far easier for the programmer to give his program the higher order of intelligence it needs in order to detect bad data or situations and subsequently notify the programmer in fairly explicit terms.

The write statement generated by the assertion statement always writes to 6.

13. Goto Statements and Labels

Although they will seldom be needed in a well written Salome program, goto statements and labels are available in Salome.

The Salome keyword '->' corresponds to the keyword "goto" in Fortran.

In Fortran, labels must be numbers (statement numbers). Fortran labels therefore have no mnemonic value. In Salome, however, label names may be any isolated string containing no blanks. Salome labels may therefore be given considerable mnemonic value.

The Salome goto statement contains the goto arrow followed by a label name. The corresponding label statement or destination of the goto consists of the label name enclosed in the special brackets: '<<' and '>>'. This convention makes labels stand out well and is used in the ADA programming language. The label statement corresponds to the 'CONTINUE' statement in Fortran.

The only keywords associated with goto and label statements are:
'->' '<<' '>>'

Ex.

```
.  
.
<< DATA-INPUT-SECTION >>
.  
.
-> DATA-INPUT-SECTION
.  
.
-> END-OF-PROGRAM
.  
.
<< END-OF-PROGRAM >>
END
```

14. Loops

Salome has a single looping statement with auxiliary loop escape and loop continue statements.

A Salome loop starts with 'DO' and ends with 'OD'. Whatever code lies between these two keywords is executed over and over again until some form of loop escape is done.

The sharp sign (#) is the basic symbol used to indicate loop escapes.

To escape from 1,2,3 or 4 surrounding loops if Boolean expression B is true, one writes:

```
# B #  
  or  
## B ##  
  or  
### B ###  
  or  
#### B ####  
Respectively
```

To escape from 1,2,3 or 4 surrounding loops unconditionally, one writes the loop escape arrows:

```
-#>  
  or  
--#>  
  or  
---#>  
  or  
----#>  
Respectively
```

To escape from more than 4 levels of loop nesting, one must use a goto and label.

While a loop escape statement breaks off execution of a loop completely, a loop continue statement breaks off execution of a single pass or iteration and then continues execution back at the beginning of the loop. The colon (:) is the basic symbol used to indicate loop continuation.

In Fortran, when one 'GOES TO' the last statement in a 'DO' loop (often a 'CONTINUE' statement), one is doing loop continuation.

To continue 1,2,3 or 4 surrounding loops if Boolean expression B is true, one writes:

```
: B :  
  or  
:: B ::  
  or  
::: B :::  
  or  
:::: B ::::  
Respectively
```

To continue 1,2,3 or 4 surrounding loops unconditionally, one writes the loop continue arrows:

```
<:-  
  or
```

```

<:--
  or
<:---
  or
<:----
Respectively

```

To continue beyond 4 levels of loop nesting, one must use a goto and label.

The unconditional loop escape and loop continue statements allow the programmer to perform some action just prior to escape or continuation.

The loop continuation constructs are seldom needed, but if they are needed, one should be careful to increment any loop index at the beginning of the loop in order to avoid an endless loop.

Ex.

```

-- MULTIPLY M X N MATRIX A TIMES N X P MATRIX B TO
-- GIVE M X P MATRIX C

```

```

-- MAKE SOME EXPLICIT ASSUMPTIONS
@ M>0 @ @ N>0 @ @ P>0 @

```

```

(-- IF THESE ASSUMPTIONS ARE NOT MADE EXPLICITLY AND ANY ONE OF THEM
TURNED OUT TO BE FALSE, NO ERROR WOULD OCCUR, BUT MATRIX C WOULD NOT BE
DEFINED AND NEITHER THE REST OF THE PROGRAM NOR THE PROGRAMMER WOULD BE
MADE IMMEDIATELY AWARE OF THIS SITUATION--)

```

```

I=0
DO I=I+1 # I>M # -- INDEX ROW OF A
  J=0
  DO J=J+1 # J>P # -- INDEX COLUMN OF B
    K=0 C(I,J)=0.
    DO K=K+1 # K>N # -- INDEX SUMMAND OF ROW A/COL B
      -- INNER PRODUCT
      C(I,J)=C(I,J)+A(I,K)*B(K,J)
    OD OD OD -- END OF MATRIX MULTIPLICATION

```

```

-- LET ARRAY A HAVE N ELEMENTS
-- ADD UP THE POSITIVE ELEMENTS OF ARRAY A
I=0 S=0.
DO I=I+1 # I>N # : A(I)<=0. : S=S+A(I) OD

```

```

-- ADD UP THE ELEMENTS OF ARRAY A HAVING INDICES BETWEEN
-- OR INCLUDING N1 AND N2

```

```

-- SOME EXPLICIT ASSUMPTIONS THAT MIGHT BE MADE ARE:

```

```

@ N1 <= N2 @ @ N1 > 0 @ @ N2 <= N @
I=N1-1 S=0.

```

```
DO I=I+1 # I > N2 # S=S+A(I) OD
(-- IF ONE DIDN'T MADE THE AFOREMENTIONED ASSUMPTIONS, ONE SHOULD BE
SOMEWHAT MORE CAREFUL IN SOLVING THE LAST PROBLEM --)
```

```
-- INITIALIZE LOOP INDEX AND ESCAPE CRITERION
IF N1<=N2 , I=N1-1 IMAX=N2 ; I=N2-1 IMAX=N1 FI
-- CHECK VALIDITY OF LOOP INDEX AND ESCAPE CRITERION
IF I<0 , I=0 FI IF IMAX>N , IMAX=N FI
S=0. DO I=I+1 # I>IMAX # S=S+A(I) OD
```

(-- ONE CAN SEE HERE THAT ALTHOUGH THE GENERAL SOLUTION TO THE LAST PROBLEM REQUIRED A LITTLE THOUGHT, THE PROGRAMMING WAS QUITE EASY --)

15. Debugging

Salome doesn't have any facility for debugging, but it does provide a mechanism for ignoring or not ignoring certain sequences of executable code at translation time.

One may prefix a '%' to a line of Salome code or one may surround multiple lines of Salome code with the delimiters '(%' and '%)'.

Salome code which is so delimited may be ignored or not ignored at translation time depending on whether the '%-Off' or '%-On' statements have been invoked, respectively. The default is %-Off.

The advantage of this facility with regard to debugging follows.

In the process of debugging, the programmer will have to insert various write statements etc. into his program in order to ascertain where the program is going wrong. This is going to be especially true if the programmer hasn't taken advantage of the explicit assumption or assertion statement.

When a Fortran programmer thinks that he has all the bugs in his program licked, he may either (1) remove the debug statements and hope that he won't have to insert them again or (2) make a comment out of each and every debug statement and hope that he won't have to change them all back to executable code.

The Salome programmer may simply insert his debug statements with % delimiting. During debugging, he may activate all or some of these debug statements using the keywords '%-On' and '%-Off'. When a Salome programmer thinks he has all the bugs in his program licked, he may simply eliminate all '%-On' strings from his program and leave all the debug statements in place for possible future use.

Ex.

SUB INTERP (N X Y XI YI) — INTERPOLATION ROUTINE

.

W (' ENTERING INTERP') .

@ N > 1 @ — ASSUME THAT THERE ARE AT LEAST TWO POINTS IN DATA

.

DX=X(1+1)-X(1)

(% IF DX = 0. , W (' DX=0. IN INTERP') EXIT /

DX < 0. , W (' DX<0. IN INTERP') EXIT FI %)

@ DX > 0. @

D=DY/DX

.

W (' EXITING INTERP') .

RET END

A FINITE DIFFERENCE PROGRAM FOR COMPUTING THE THERMOELASTIC-
PLASTIC RESPONSE OF LINED GUN BARRELS

John D. Vasilakis
U.S. Army Armament Research and Development Command
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189

ABSTRACT. A finite difference computer program for computing the thermoelastic-plastic response of multilayered cylinders due to repeated firing loads was discussed at the 27th Conference of Army Mathematicians. The multilayered cylinder is a representation of a lined gun barrel. The program can accommodate several layers and can compute the transient temperatures and/or the stresses. It has been upgraded to include an initial program which computes heat transfer coefficients, pressures and gas temperatures in the firing cycle for input to the main program. The effect of contact resistance between layers is now included. Results are shown for the behavior of a TZM liner in a steel tube.

1. INTRODUCTION. This paper describes a finite difference computer program for investigating the response of multilayered gun barrels subject to some firing cycle. Results, typical of which the program is capable of generating, are presented for a tube model which has a TZM liner and a steel jacket. The application is to a large caliber weapon but the program can be used for small caliber also. The program was written to coincide with a development program which is examining the feasibility of fabricating and firing multilayered gun tubes. One of the main factors limiting the life of gun tubes is the excessive wear and erosion which occurs especially at the forcing cone area of the gun tube. The experimental program, which has shown success for 20 mm weapons, is to insert liners fabricated from refractory materials into the forcing cone area of the gun tube. Since refractory materials have high melting points, there is a strong indication that they will experience less wear and thus increase the life of the weapon.

Earlier versions of the computer program have been used to describe other behavior (refs. 1-3). Preliminary work on the current problem was presented in reference (3). That work has been improved by inputting the thermo-physical properties as functions of temperature and by allowing contact resistance between layers. The boundary conditions have also been improved so they are now generated for the current problem at hand, i.e., for specific configuration and bore material whereas previously they were empirically generated for another system and simply used in the program as typical input.

The computer program consists of three parts which can be run as a single program or as three separate stand alone programs. The first is an internal ballistics program which generates the boundary conditions, i.e., heat transfer coefficients, pressures, and gas temperatures as a function of time for a single firing pulse, for input to the next two programs. The next

program section computes the transient temperatures due to some firing cycle and can be used to either show the thermal response of the system over several firing cycles, indicating the temperature buildup and/or the temperatures can be used as input to the third program section for the computation of stresses. This can be done as the thermomechanical program is treated as uncoupled.

Results from each of the three sections using a TZM liner/steel jacket configuration for a 105 mm large caliber weapon system are presented.

2. PROCEDURE. The equations used to describe the behavior are first discussed in subsection (a) and the numerical work in (b).

a. Theory. The partial differential equation for describing the axisymmetric transient temperature distribution in multilayered cylinders is given by, for layer L,

$$\frac{1}{r} \frac{\partial}{\partial r} (k^L(T)r \frac{\partial T^L}{\partial r}) = c^L(T)\rho^L(T) \frac{\partial T^L}{\partial t} \quad (1)$$

where r represents the radial distance, T the temperature, and t the time. The thermal conductivity, specific heat, and density are given by k, c, and ρ respectively. These properties are assumed to be functions of temperature. Axial effects are ignored in the program. The geometry is shown in Figure 1.

The initial condition is given by

$$T(r,0) = T_0 \quad (2)$$

where T_0 would normally represent some ambient temperature. A temperature other than ambient, say due to some environmental condition, could also be used. The boundary conditions are of the type

$$k(T) \frac{\partial T}{\partial r} - h(T-T_g) = -g \quad (3)$$

where h is convection type heat transfer coefficient, g would represent some heat input if it existed, and T_g is the temperature of the propellant gases when the boundary condition is applied on the inside or bore diameter and the ambient temperature when applied on the outside surface of the gun tube.

The thermo-physical properties are made dimensionless with respect to their respective values for steel at the ambient temperature. The temperatures are made dimensionless with respect to the maximum gas temperature achieved during the interior ballistic cycle. The dimensionless time is defined by

$$\tau = \frac{k_0 t}{\rho_0 c_0 b^2} \quad (4)$$

where k_0 , ρ_0 , and c_0 are the values for steel as mentioned previously. In the boundary conditions,

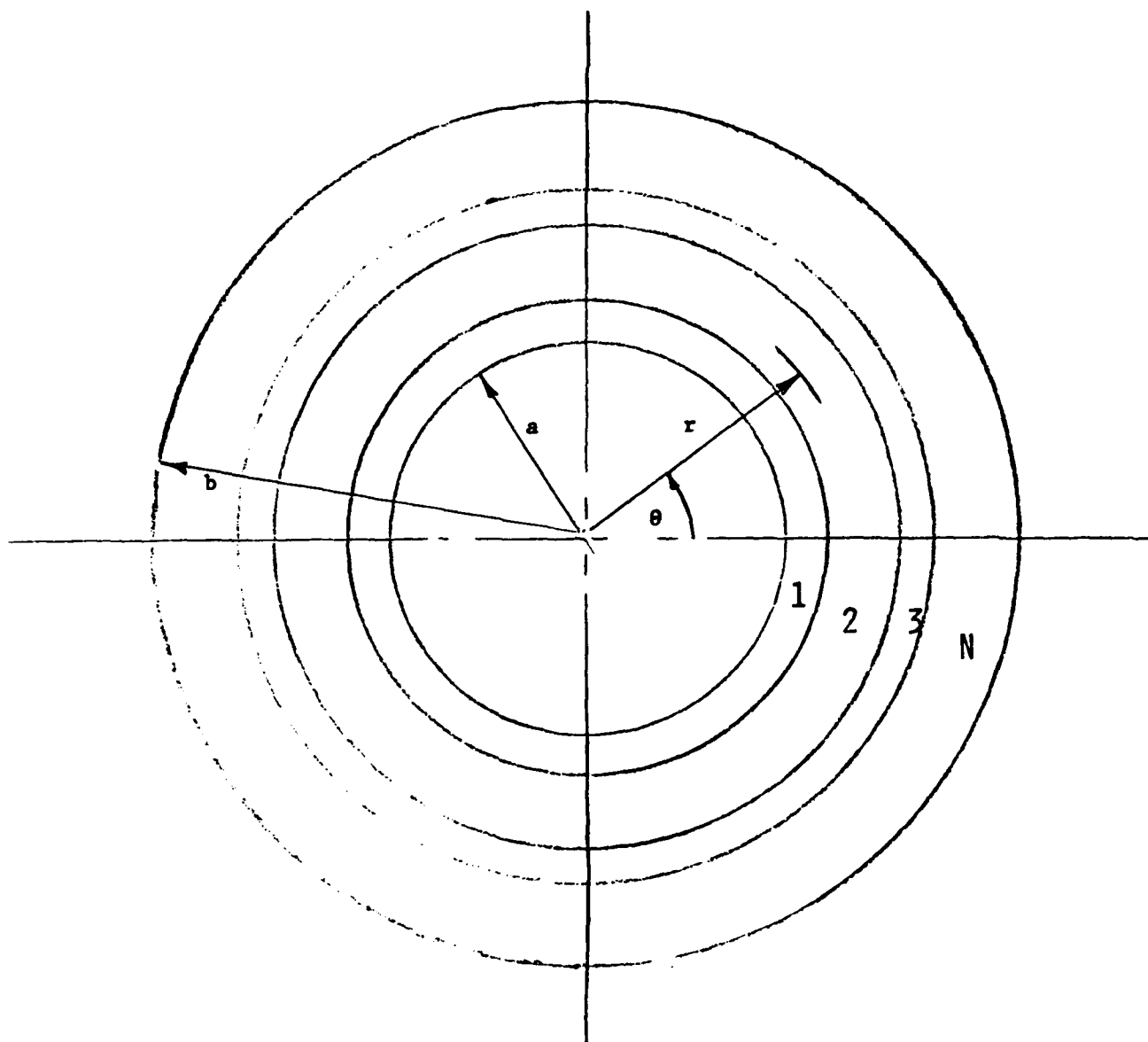


FIGURE 1. TYPICAL MULTI-LAYERED GEOMETRY

$$\hat{h} = \frac{hb}{k} \quad (5)$$

becomes the Nusslet number.

Also required are continuity conditions between the concentric cylinders. Normally one requires continuity of temperature and heat flux. However, contact resistance does exist between surfaces and it was decided to include this effect here. The resistance results from the true nature of surfaces (ref. 4). Conduction occurs at the discrete points of contact between the surfaces and is therefore a function of pressure, surface conditions, fluids in the voids, etc. It is treated here as a thin layer resisting the flow of heat. For the finite difference formulation, there is a jump in temperature at node 1

$$T_1^L = T_1^{L+1} + \Delta T \quad (6)$$

while the heat flux

$$\frac{q}{A} = k^L(T) \frac{T_{i-1}^L - T_1^L}{\Delta r} = h_c \Delta T = k^{L+1}(T) \frac{T_1^{L+1} - T_{i+1}^{L+1}}{\Delta r} \quad (7)$$

remains constant as one passes across the layer from cylinder L to cylinder L+1. Equations (6) and (7) allow the computation of ΔT and T_1^{L+1} .

The use of finite difference equations to solve the thermo-elastic-plastic stress problem requires expressing the equilibrium equation and the equation of compatibility at each node at which the finite difference equations are desired. The Prandtl-Reuss flow rule is used to eliminate the incremental stresses so that what results is a matrix for evaluating the incremental radial and tangential strains at each node. The required equations follow, written in dimensionless form. The problem is treated as plane strain.

The equation of equilibrium is written

$$\frac{\partial \sigma_r}{\partial r} + \frac{\sigma_r - \sigma_\theta}{r} = 0 \quad (8)$$

where

$\sigma_r (= \frac{\sigma_r}{\sigma_0})$ is the dimensionless radial stress

$\sigma_\theta (= \frac{\sigma_\theta}{\sigma_0})$ is the dimensionless tangential stress

and σ_0 is the yield stress in tension, and the compatibility equation

$$\frac{\partial \epsilon_\theta}{\partial r} + \frac{\epsilon_\theta - \epsilon_r}{r} = 0 \quad (9)$$

where

$\epsilon_\theta (= E \frac{\epsilon_\theta}{\sigma_0})$ is dimensionless tangential strain

$\epsilon_r (= E \frac{\epsilon_r}{\sigma_0})$ is dimensionless radial strain

and σ_0/E is yield strain in tension when E is Young's Modulus. The compressibility of the material is expressed by

$$\epsilon = \alpha T + \frac{\sigma}{3K} \quad (10)$$

$\epsilon = \frac{1}{3} (\epsilon_r + \epsilon_\theta)$ is mean strain

$\sigma = \frac{1}{3} (\sigma_r + \sigma_\theta + \sigma_z)$ is mean stress

$K (= \frac{K}{\sigma_0})$ is dimensionless bulk modulus

$\alpha (= \alpha T_1)$ is dimensionless coefficient of thermal expansion

and

$\epsilon_z = 0$ for plane strain

Traction free boundary conditions are always used in the outside radius and on the bore when only thermal stresses are required. When mechanical loads are desired, the pressure pulse is applied to the bore.

It was desirable to write the finite difference equations in terms of strain alone, hence, the stresses in the equations of equilibrium had to be expressed in terms of the strains. This was accomplished by modifying a plastic stress-strain matrix (ref. 5) which was derived by inverting the Prandtl-Reuss equations. The inverted Prandtl-Reuss equation is

$$\{d\sigma\} = [D^P] \{d\epsilon\} - \frac{E \alpha dT}{(1-2\nu) \sigma_0} \{1\} \quad (11)$$

where the stress vector is $\{d\sigma\} = \{d\sigma_r, d\sigma_\theta, d\sigma_z\}^T$, the strain vector $\{d\epsilon\} = \{d\epsilon_r, d\epsilon_\theta, 0\}^T$, and $\{1\}$ represents a unit vector. The plastic stress-strain matrix $[D^P]$ is given by

$$[D^P] = \frac{1}{1+\nu} \begin{bmatrix} \frac{1-\nu}{1-2\nu} - \frac{\sigma_r'^2}{S} & & & \\ & \frac{\nu}{1-2\nu} - \frac{\sigma_r'\sigma_\theta'}{S} & \frac{1-\nu}{1-2\nu} - \frac{\sigma_\theta'^2}{S} & \\ & \frac{\nu}{1-2\nu} - \frac{\sigma_r'\sigma_z'}{S} & \frac{\nu}{1-2\nu} - \frac{\sigma_\theta'\sigma_z'}{S} & \frac{1-\nu}{1-2\nu} - \frac{\sigma_z'^2}{S} \\ & & & \end{bmatrix} \quad \text{SYMMETRIC} \quad (12)$$

The primed stresses are deviatoric stresses,

$$\sigma_i' = \sigma_i - \frac{1}{3} \sum \sigma_i \quad i = r, \theta, z \quad (13)$$

At each node during a computation, the von Mises' yield criterion

$$\frac{1}{2} [(\sigma_r - \sigma_\theta)^2 + (\sigma_\theta - \sigma_r)^2 + (\sigma_z - \sigma_r)^2] = 1 \quad (14)$$

is checked to see if plastic deformation has progressed to that node. If not, the stresses remain elastic and can still be computed using Eq. (12) by setting the deviatoric stresses equal to zero. The matrix $[D^P]$ then becomes the same matrix as would exist if linear elastic behavior had been assumed. The quantity S is given by

$$S = \frac{2}{3} \bar{\sigma}^2 \left(1 + \frac{H'}{3G}\right) \quad (15)$$

where

$$\bar{\sigma} = \frac{3}{2} \sigma_{ij}' \sigma_{ij}' = \frac{3}{2} (\sigma_r'^2 + \sigma_\theta'^2 + \sigma_z'^2) \quad (16)$$

is the equivalent stress and

$$H' = \frac{d\sigma}{d\epsilon_p} \quad (17)$$

is the slope of the equivalent stress/equivalent plastic strain curve and is a measure of hardening. The increment in equivalent plastic strain is given by

$$d\epsilon_p = \frac{2}{3} d\epsilon_{ij}^p d\epsilon_{ij}^p \quad (18)$$

b. Boundary Conditions. The boundary conditions that are used as input for the calculation of temperatures and of stresses are generated using a computer program based on reference (6). In that paper, the burning of a specific propellant for the purpose of firing a projectile from a gun tube is modeled. The equations used are based on Corner's work (ref. 7) and represent a first order interior ballistics solution. Lagrange's approximation is assumed, i.e., the velocity of the gas at any instant increases linearly with distance along the bore from zero at the breech to the full shot velocity at the base of the projectile. Exponential decay is assumed during the blowdown cycle, i.e., after the projectile has left the gun tube. Based upon the rate equations and heat balances involved, the heat transfer coefficients, the pressure pulse, and the gas temperature can be found as a function of time during the firing cycle. The quantity of heat that goes into the heating of the gun tube can be computed. The bore surface can be specified to be the liner material. Figure 2 shows the output of this program for a gun tube with steel at the bore surface.

c. Numerical Procedure. The Crank-Nicolson representation for finite differences of the partial differential equation governing the temperatures in time is (ref. 1)

$$\begin{aligned}
 & [(a+i\Delta r)k_{i+1/2,n+1/2}]T_{i+1,n+1} + [-(a+i\Delta r)k_{i+1/2,n+1/2} \\
 & -(a+(i-1)\Delta r)k_{i-1/2,n+1/2}\rho_{i,n+1/2}(\frac{2\Delta r^2}{\Delta t})(a+(i-1/2)\Delta r)]T_{i,n+1} \\
 & + [(a+(i-1)\Delta r)k_{i-1/2,n+1/2}]T_{i-1,n+1} = [-(a+i\Delta r)k_{i+1/2,n+1/2}]T_{i+1,n} + \\
 & + [(a+i\Delta r)k_{i+1/2,n+1/2} + (a+(i-1)\Delta r)k_{i-1/2,n+1/2} - \\
 & c_{i,n+1/2}\rho_{i,n+1/2}(\frac{2\Delta r^2}{\Delta t})(a+(i-1/2)\Delta r)]T_{i,n} + [-(a+(i-1)\Delta r)k_{i-1/2,n+1/2}]T_{i-1,n}
 \end{aligned}
 \tag{19}$$

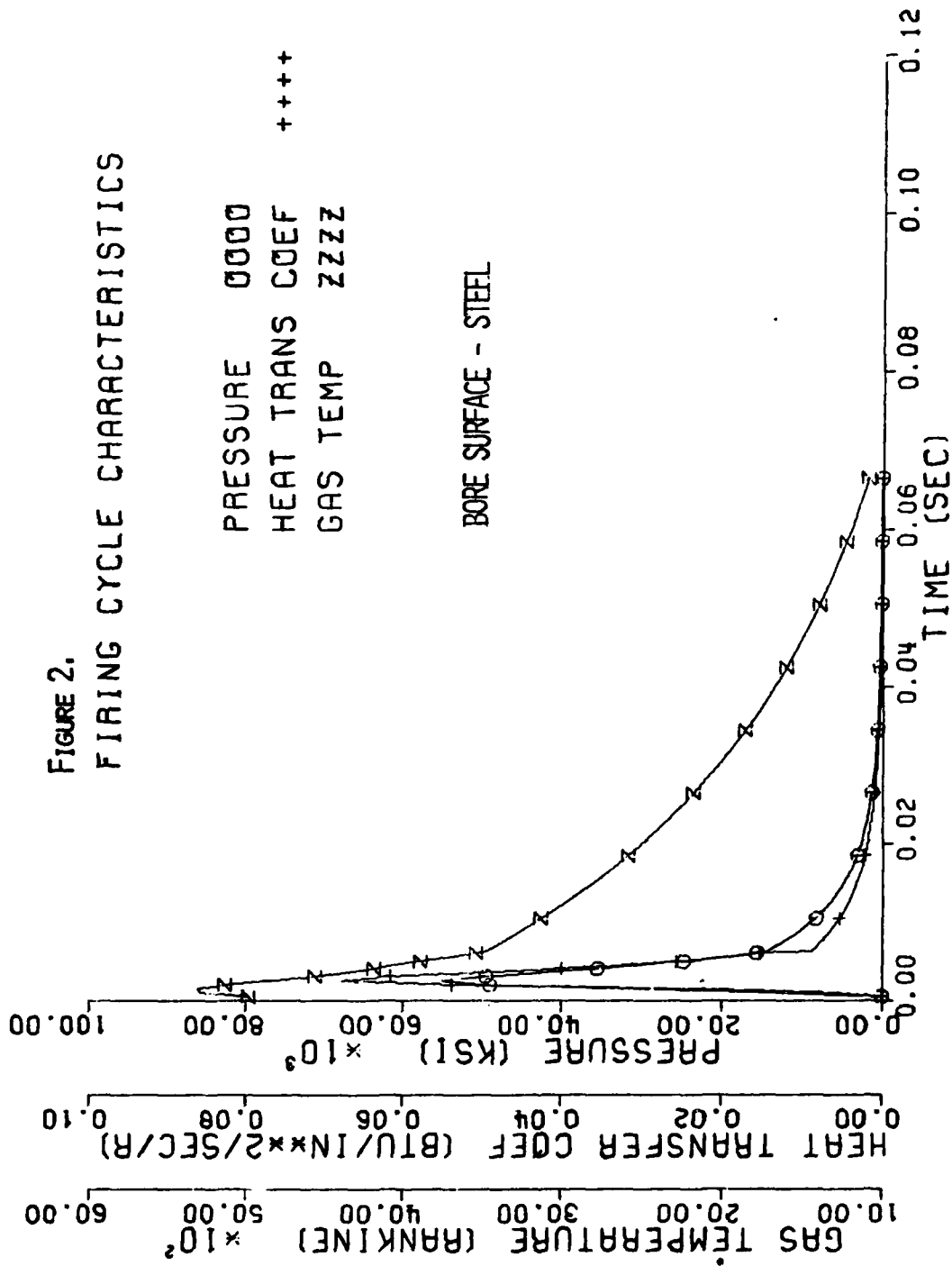
The equation is solved twice:

1. At $n+1/2$ step, allowing k, ρ, c etc. to take on the values at $t = n$ step.
2. The new temperatures are then used to evaluate k, c, ρ at $n+1/2$ step and the set of equations re-evaluated for the temperatures at the $n+1$ step.

The computed temperature distributions at each full time step are saved on disk and eventually called in when required by the stress program.

The finite difference equations are within any layer.

FIGURE 2.
FIRING CYCLE CHARACTERISTICS



Compatibility:

$$\begin{aligned}
 & -r_1 \Delta \epsilon_{\theta_{i-1}} + (2r_1 - r_{i-1}) \Delta \epsilon_{\theta_i} - (r_1 - r_{i-1}) \Delta \epsilon_{r_i} = \\
 & -r_1 (\epsilon_{\theta_i} - \epsilon_{\theta_{i-1}}) - (r_1 - r_{i-1}) (\epsilon_{\theta_i} - \epsilon_{\theta_i}) \quad (20)
 \end{aligned}$$

Equilibrium:

$$\begin{aligned}
 & -r_1 \Delta \sigma_{r_{i-1}} - (r_1 - r_{i-1}) \Delta \sigma_{\theta_i} + (2r_1 - r_{i-1}) \Delta \sigma_{r_i} \\
 & -r_1 (\sigma_{r_i} - \sigma_{r_{i-1}}) - (r_1 - r_{i-1}) (\sigma_{r_i} - \sigma_{\theta_i}) \quad (21)
 \end{aligned}$$

Substituting the Prandtl-Reuss equations into that of equilibrium

$$\begin{aligned}
 & -r_1 D(r, \theta) \Delta \epsilon_{\theta_{i-1}} - r_1 D(r, r) \Delta \epsilon_{r_{i-1}} + [-(r_1 - r_{i-1}) D(\theta, \theta) + (2r_1 - r_{i-1}) D(r, \theta)] \Delta \epsilon_{\theta_i} \\
 & + [-(r_1 - r_{i-1}) D(\theta, r) + (2r_1 - r_{i-1}) D(r, r)] \Delta \epsilon_{r_i} \\
 & r_1 [\sigma_{r_{i-1}} - \sigma_{r_i}] + (r_1 - r_{i-1}) (\sigma_{\theta_i} - \sigma_{r_i}) + r_1 \frac{E\alpha}{1-2\nu} [\Delta T_i - \Delta T_{i-1}] \quad (22)
 \end{aligned}$$

Equations (20) and (21) are in backward difference equations. The actual computations are performed by averaging backward and forward difference schemes. At the interface between cylinders, continuity of the radial stress and radial displacement is specified and on the boundary, $i = 1$,

$$D(r, \theta) \Delta \epsilon_{\theta_1} + D(r, r) \Delta \epsilon_{r_1} = \frac{E\alpha \Delta T_1}{1-2\nu} - \Delta p_1 \quad (23)$$

where Δp_1 represents a pressure increment at the bore or inside diameter.

The solution procedure for the transient temperature problem is as follows. The temperature problem is solved, and the temperature distributions at their computation times are stored on disk. These distributions are called into the thermo-elastic-plastic stress program one at a time. The corresponding thermal stresses are calculated and each node checked to see if the yield criterion is satisfied. If not, the problem is still assumed to be elastic, a new temperature distribution is called in, and new stress increments calculated. The stresses are updated, and the yield criterion checked again. When the stresses at a point are found to satisfy the yield criterion the node is identified, and the stress increments at that node from the next set of temperatures are computed using the Prandtl-Reuss equation or $[D^P]$ matrix identified earlier. This procedure is continued with new sets of temperature called in and with the tracking of the elastic-plastic boundary with time.

The mechanical properties are evaluated at the existing temperatures. However, the yield stress has not yet been incorporated as a function of temperature in the program.

d. Mechanical and Thermophysical Properties. The properties used in the calculations were found in reference (8). The nominal values are given in Table 1. It is always one of the more difficult tasks to find properties as functions of temperature. The steel properties used were those of 4340 and 4150. "Gun Steel" is typically 4340 or a modification thereof. The thermo-physical properties for TZM used were those for those for molybdenum itself since they were readily available as functions of temperature and the same properties for TZM, only given at one or two specified temperatures tended to fall on or near the same property for molybdenum.

TABLE 1. ROOM TEMPERATURE PROPERTIES

	k	c	ρ	α	E	σ_{yp}	ν
	BTU/#in°F	BTU/#°F	#/in ³	in/in°F	Psi	Psi	
Steel	5.01×10^{-4}	.105	.289	6.2×10^{-6}	30×10^6	160×10^3	.3
TZM	1.87×10^{-4}	.06	.369	3.0×10^{-6}	45×10^6	130×10^3	.314

3. RESULTS. The interior ballistic code was first run to set up the input data (heat transfer for coefficients and gas temperatures during firing cycle) for the position of the program which computes the transient temperature distribution and pressure time curve for the mechanical load contribution to the stress part of the program. Two data sets were established, one for steel at the bore and one for TZM at the bore. Figure 2 showed the results for steel. The interior ballistic code computes these results at several stations along the tube, but only the section at which the maximum pressure was generated was considered at this time. The temperature portion of the program serves two separate purposes. The program can be run over several firing pulses based on some specific firing cycle. This will show the buildup in temperature during the firing. It can also be run to provide input data in the form of temperature distributions throughout the wall of the tube for specific times during the firing cycle. This data set is then used in the stress program for the computation of thermal stresses or thermo-mechanical stresses when the pressure-time curve is also applied. There were four types of material problems considered, a single (monobloc) steel tube with constant properties, a TZM liner/steel jacket with constant properties in each cylinder and a TZM liner/steel jacket with temperature dependent properties. Results from some of these cases are presented below.

Figure 3 shows the change of the bore temperature with time over four firing cycles. The configuration is the TZM liner/steel jacket with temperature dependent properties. The firing cycle depicted represents a projectile being fired at the rate of four rounds per minute. The temperature buildup at the bore can easily be seen. Figure 4 shows the temperature response at the bore for a monobloc steel tube with temperature dependent properties. This is shown on an expanded scale and represents the thermal response used on one of the data sets for the stresses. When these data sets

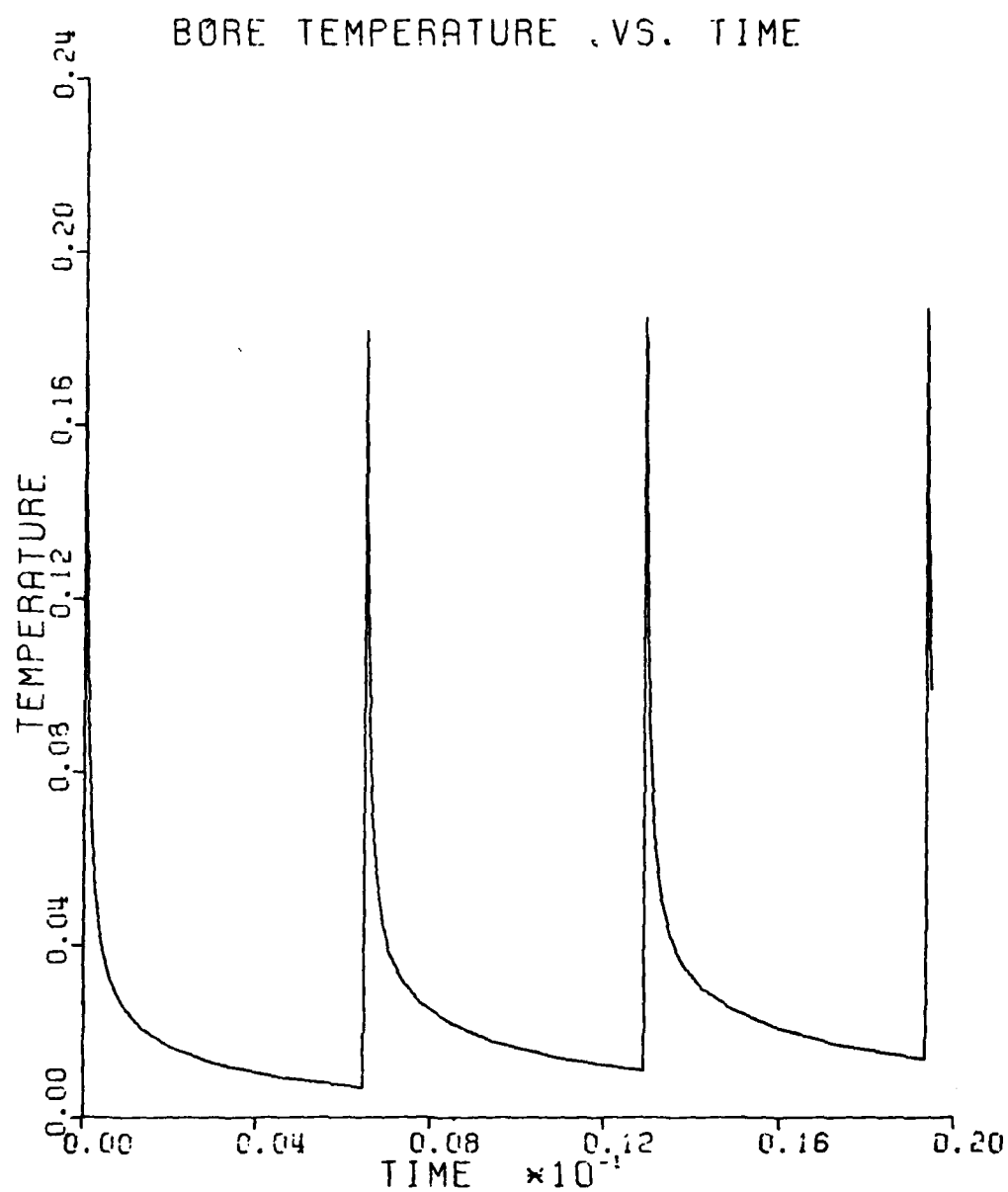


FIGURE 3. BORE TEMPERATURE RESPONSE OVER SEVERAL FIRING CYCLES.

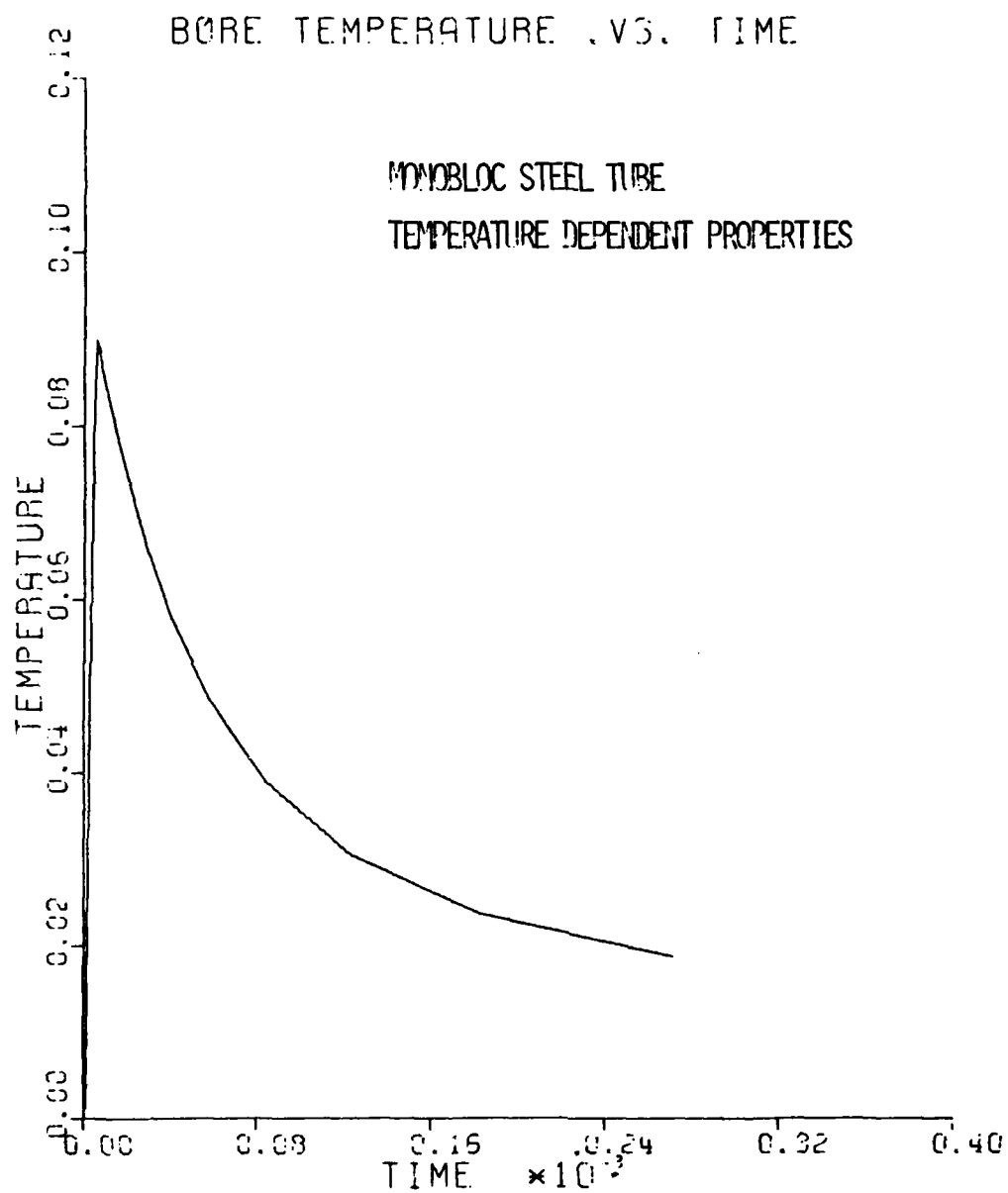


FIGURE 4. BORE TEMPERATURE VS TIME.
RESPONSE FOR INPUT TO STRESS PROGRAM.

for stresses are established, a finer time increment is used over that which simply computes the thermal response for temperature buildup. The finer time increment decreases the temperature difference, ΔT , from time step to time step which is used in the stress program. This helps in approaching yield in the stress program even though the temperature difference is further divided near yielding and after yielding has begun. When only the temperatures themselves are desired, it was previously shown (ref. 3) that larger time increments can be used. As the finite difference program for temperatures is implicit, the time increment between pulses is increased at a rapid rate until the next firing pulse comes along. It took approximately 90 time steps to complete the temperature response to the four cycles.

When investigating the effect of contact resistance, it becomes obvious why many papers ignore it. The difficulty is not with the computation, but rather the uncertainty of the physical constant to use in the evaluation. The property h described earlier, is treated as a constant here but in reality would be a function of pressure, temperature, the roughness of the surfaces in contact, the hardness of the materials involved, etc. One would actually need the true area of contact as opposed to the apparent area and how this changed in time. Table 2 shows the effect of varying h on the bore temperature for five firing pulses. The h is dimensionless. The table shows that when $h = 1000$, the system is equivalent to having no resistance to heat flow and when $h = .00001$, the effect is equivalent to zero heat flow at the interface. As can be seen by converting some of the values from reference (4), one can get coefficients which result in measurable effects. To better show the effect, the resulting temperature distribution as a function of radius as shown in Figure 5 for $h = 1$. The second, third, and fourth pulse are displaced from the first for clarity of viewing. The bore temperature increases substantially compared to zero contact resistance (as can also be seen from Table 2). The temperature jump also increases substantially. The line indicating the jump should be a vertical line from the point on the inner cylinder. The fact that it is not is due to the plotting routine. A final remark on this section would be that with the uncertainty in computing or experimentally determining an actual h for a system, Table 2 shows little difference between an $h = 1000$ and $h = 100$ as a threshold for zero contact resistance and again little difference between $h = .1$ and $h = .00001$ for an insulating barrier.

TABLE 2. EFFECT OF h ON BORE TEMPERATURE

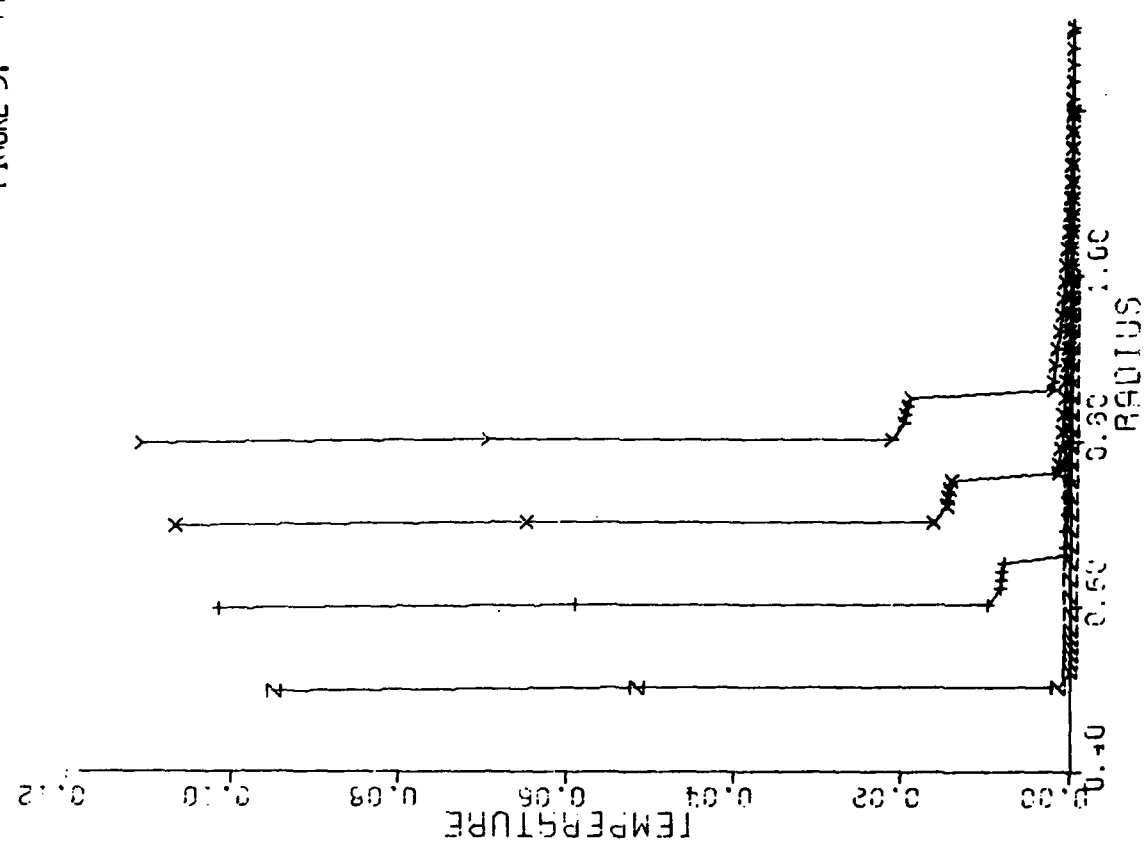
		1000	100	1	.1	.00001
Temp	Pulse 1	0	0	0	0	0
	Pulse 2	.00352	.00356	.00837	.00999	.01021
	Pulse 3	.00580	.00582	.01493	.01952	.02020
	Pulse 4	.00758	.00756	.02013	.02861	.02998
	Pulse 5	.00904	.00901	.02424	.03728	.03596

FIGURE 5. TEMPERATURE VS RADIUS
CONTACT RESISTANCE EFFECTS

H = 1.0

1ST PULSE Z
2ND PULSE +
3RD PULSE X
4TH PULSE Y

2ND, 3RD AND 4TH PULSE ARE
DISPLACED FROM 1ST FOR VIEWING



The remaining results relate to the output from the stress portion of the program. Figure 6 shows the effect of including temperature dependence of material properties in the analysis. The variation of radial stress and tangential stress across the wall thickness are shown. There is little effect except at the peak tangential stress at the bore which is about three to four percent higher for a tube with temperature dependent material properties. This represents a stress difference of about 5000 psi.

The stress results being shown are only the response to the first firing pulse. Hence, the stresses are mainly due to a steep thermal gradient at the bore of the tube. If inelastic response does not occur, the stress response would not change significantly from pulse to pulse as the mechanical load vanishes and there remains only a slight thermal gradient throughout the tube wall. Depending on the configuration of the system, the firing rate, etc., the rise in bore temperature before the next round is fired is small, $\sim 3^\circ\text{F}$ to 9°F .

Figure 7 shows the variation in tangential stress versus radius for the multi-cylinder configuration, TZM/steel, with the temperature dependent properties at the time when the internal pressure peaks. The mechanical loading due to the pressure-time curve alone is shown as is the thermal stress distribution due to the temperature distribution. The combined effects are also shown. These combined effects are not arrived at by adding the separate ones but are recomputed. This is important, especially in the case shown, because the mechanical load alone was sufficient to cause the inelastic deformation at the bore. While this inelastic zone was concentrated at the bore and there was little depth of plastic zone penetration into the wall, it is still incorrect to linearly superimpose solutions by adding the separate stress distributions. Under the combined loads, the solution remained elastic throughout. Figure 8 shows the results from the same problem at the time when the thermal stresses peak.

4. SUMMARY. The computer program described is capable of predicting the thermal response and the thermo-elastic-plastic response of liner/jacket gun tube designs with temperature dependent thermo-physical properties. Realistic input to this problem is generated using an interior ballistics code. While only a two cylinder system is described, allowance is made for up to five cylinders.

Improvements can always be made and in this case, the following could be included. By using a variable space increment in the thermal program, the effect of a deposited thin layer on the bore surface can be investigated. Initial stresses due to interference should be incorporated as should a temperature dependent yield stress.

ACKNOWLEDGEMENT. The author would like to thank Pat Vottis for allowing the use of his program to generate proper boundary conditions.

FIGURE 6. STRESS VS RADIUS
TEMPERATURE DEPENDENCE
EFFECTS FOR STEEL TUBE

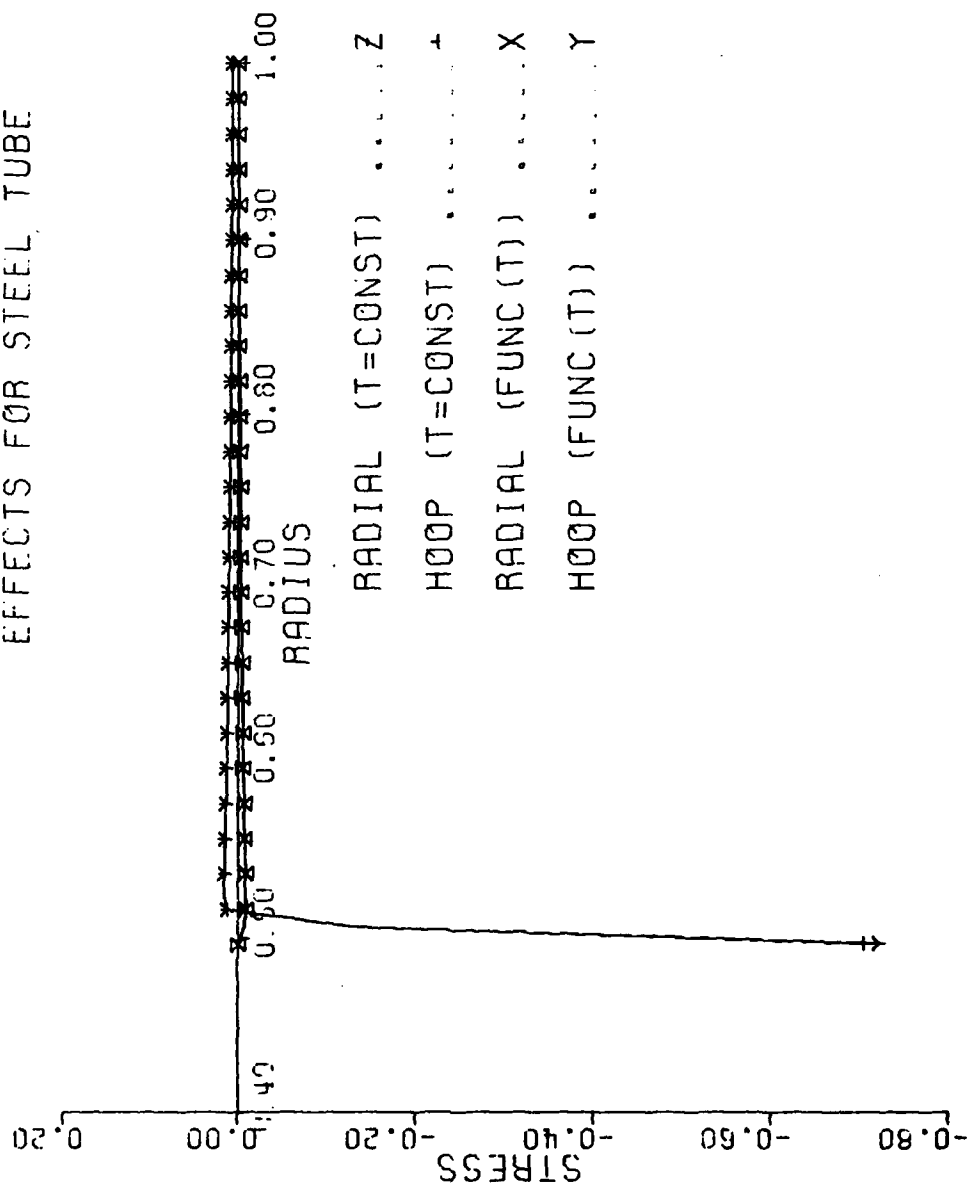
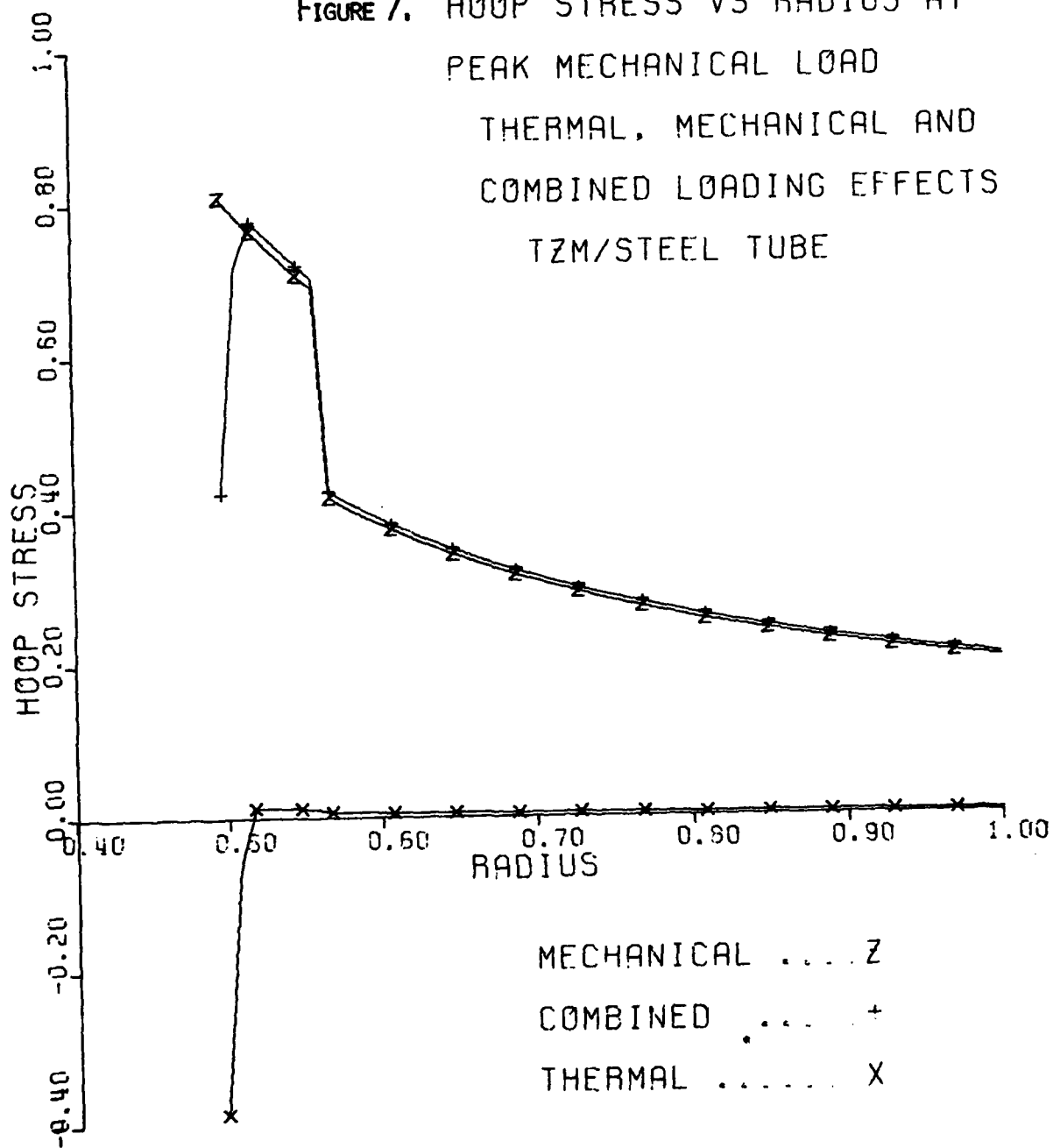
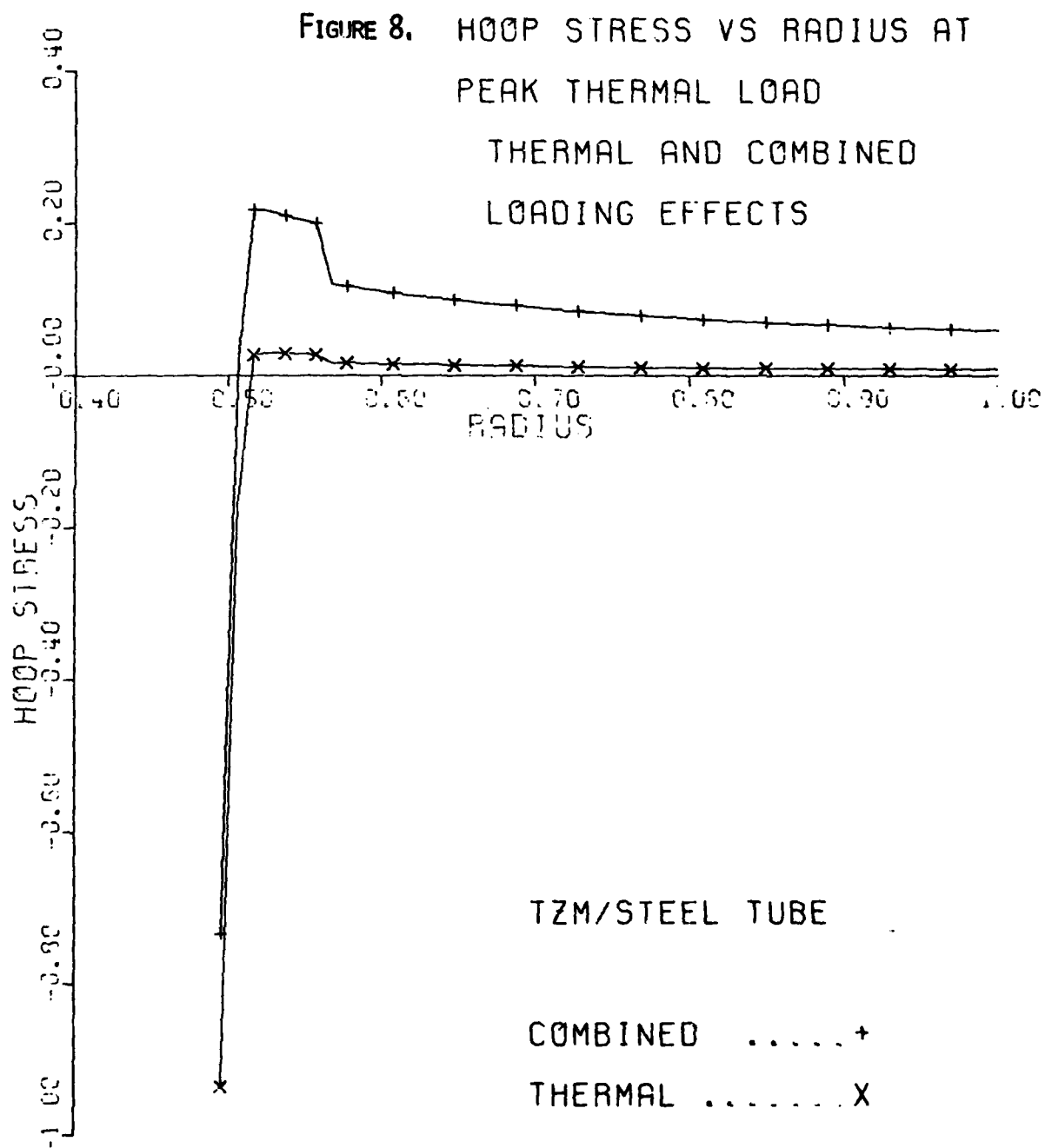


FIGURE 7. HOOP STRESS VS RADIUS AT
PEAK MECHANICAL LOAD
THERMAL, MECHANICAL AND
COMBINED LOADING EFFECTS
TZM/STEEL TUBE





REFERENCES

1. Vasilakis, J. D., "Temperatures and Stresses Due to Quenching of Hollow Cylinders," Transactions of the 24th Conference of Army Mathematicians, ARO Report 79-1, January 1979.
2. Vasilakis, J. D. and Chen, P. C. T., "Thermo-Elastic-Plastic Stresses on Hollow Cylinders Due to Quenching," Transactions of the 25th Conference of Army Mathematicians, ARO Report 80-1, January 1980.
3. Vasilakis, J. D., "Thermo-Elastic-Plastic Stresses in Multi-Layered Cylinders," Transactions of the 20th Conference of Army Mathematicians, ARO Report 82-1, January 1982.
4. Fenech, H. and Rohsenow, W. M., "Prediction of Thermal Conductance of Metallic Surfaces in Contact," Journal of Heat Transfer, February 1963, p. 15-24.
5. Yamada, Y., Yoshimura, N., and Sakuri, T., "Plastic Stress-Strain Matrix and Its Application for the Solution of Elastic-Plastic Problems by the Finite Element Method," International Journal of Mechanical Sciences, 1968, Vol. 10, pp. 343-354.
6. Vottis, P. M., "Digital Computer Simulation of the Interior Ballistic Process in Guns," WVT-6615, Watervliet Arsenal, Watervliet, NY, October 1966.
7. Corner, J., Theory of the Interior Ballistics of Guns, John Wiley and Sons, Inc., New York, 1950.
8. Aerospace Structural Metals Handbook, AFML-TR-68-115.

ACCURATE COMPUTER ARITHMETIC FOR SCIENTIFIC COMPUTATION

L. B. Ral.
Mathematics Research Center
University of Wisconsin-Madison

ABSTRACT. A basic requirement of scientific and engineering computing should be that the numerical results are obtained as accurately as possible. On existing computers, however, it is well-known that arithmetic operations are not always performed to this standard. For example, the UNIVAC 1100 series execute most single precision floating-point operations with a relative error of 2^{-27} ; however, one gets

$$134217728.0 - 134217727.0 = 2.0,$$

a result which is in error by 100%, even though the arguments are exactly representable. Similar errors are made by other machines in their floating-point operations. This situation can be avoided, and high accuracy obtained in final results, if computer arithmetic is done according to the general theory developed over the last ten years by U. Kulisch (working at MRC, IBM, and the University of Karlsruhe) and his coworkers. This accurate (or controlled) arithmetic is based on the ordinary operations $+$, $-$, $*$, $/$ augmented by ∇ (downward rounding), Δ (upward rounding), \square (antisymmetric rounding), and, for accurate numerical linear algebra, a scalar product $*$ of maximum precision. The operations of accurate arithmetic can be implemented easily on a microcomputer or a computer with microprogrammable arithmetic operations. In addition, for accurate final results, a compiler is necessary which will select the appropriate operations. A language of this type (PASCAL-SC) has been developed in Germany. Along with accurate real arithmetic, this compiler provides accurate complex arithmetic, real and complex interval arithmetic, and vector and matrix arithmetic over these data types. Some features of PASCAL-SC related to scientific and engineering computation will be described. In most cases, the operations of accurate arithmetic are performed at the same speed as ordinary (uncontrolled) floating-point arithmetic.

1. IMPORTANCE OF ACCURACY. Scientific computation, in contrast to other applications of electronic digital computers, requires the performance of large numbers of arithmetic operations in most cases. Although computers were originally brought forth to do "number crunching" on scientific and engineering problems, this area has faded in importance, so that most computers today are used to manipulate data files most of the time, with only the most elementary arithmetic being done, such as for payrolls and other accounting purposes. The evolution of computers has followed, by and large, adaptation to tasks for which the sales of machines are the greatest. In particular, accuracy has been a minor consideration in the design of arithmetic units, and present-day machines show no improvement in this area, or even a deficiency as compared to earlier machines. The example given in the Abstract is not at all isolated; similar defects can be found in other models of machines in wide use today. While such "glitches" are extremely rare, and can be detected by programs which have enough internal checking of consistency of results, it is probably only a matter of time until some disaster is traceable to an error in computer arithmetic inherent in the design of the machine. In addition to the scientific and engineering computing community, it would appear that insurance

Research sponsored by the U. S. Army under Contract No. DAAG29-80-C-0041.

companies, as well as the public in general, should be interested in improvement of the accuracy with which computers perform the basic arithmetic operations.

The type of inaccuracy considered here is not the inevitable error in using one of the two floating-point neighbors of a real number to represent it; this error is usually very small. What is under discussion is the *unnecessary failure* of a computer to produce such a neighboring number as the result of an arithmetic operation, and the lack of controllability of the rounding of results to the desired neighbor (closest, greater, or lesser). The inaccurate, uncontrolled floating point arithmetic one finds at present has a harmful effect on the reliability of computed results, and it should be replaced by accurate, controlled arithmetic.

The question of increase of accuracy in computer arithmetic is more subtle than may appear at first sight. For example, the arithmetic unit of the UNIVAC 1100 could have been designed in such a way that errors of the type cited would not occur [29], [31], and the resulting change in price of the machine would have been negligible. It would seem then that if users set standards for arithmetic accuracy, then the manufacturers could respond by supplying the desired product without much disruption of the overall architecture of the machine, whether or not number crunching is viewed as its primary purpose. However, the following questions arise:

1°. What are explicit standards for arithmetic accuracy?

2°. How does one build an arithmetic unit to implement such standards?

There is even a further question which must be addressed in scientific and engineering computation:

3°. How can one develop software to take advantage of accurate arithmetic to produce results which are as accurate as possible, and give a reliable indication of their accuracy?

This last question is of particular importance at the present time, since most people use software developed by others, which is written in higher-level languages and run on a variety of computers with different basic accuracies. The overall problem of accuracy, therefore, involves both hardware and software, and may appear to be intractable in general. However, a fundamental theory of computer arithmetic has recently been developed by U. Kulisch [9], [10], [11], [12], and his coworkers, and is described in detail in the book by Kulisch and Miranker [13]. This theory provides a guide to the construction of accurate hardware and software, and the latter has been implemented for a microcomputer [14]. A brief and incomplete discussion of these developments will be given below; the book [13] should be consulted for a fuller treatment. The main point is that it is possible to answer the questions posed above on the basis of a rigorous mathematical theory instead of *ad hoc* reasoning applied to special cases.

Unfortunately, there is an attitude of satisfaction with the *status quo* among certain members of the scientific and engineering computing community. Since computations cannot be performed exactly in general, a tolerance of unnecessary errors in arithmetic has developed. One hears comments such as, "My data aren't very good anyway, and I only need a couple of decimal places in the results." An answer to this is the calculation cited above, in which the data are exact, but the result does not contain *any* accurate digits. Another widely circulated idea is that higher

precision is a solution to the accuracy problem. One can compare the number of digits to which single and double precision computations agree, for example. (People who advocate this idea seldom carry it out: They simply use the highest precision available at low cost.) It is true that the number of accurate digits are usually increased by going to higher precision, but usually at the cost of an even greater increase in the (unknown) number of inaccurate digits. Poorly designed arithmetic units will also make mistakes in double or higher precision, thus obviating this approach. It is the position of this paper that confidence in the result of numerical computations should be based on reason, not faith. Furthermore, manufacturers should be required to deliver machines capable of accurate computer arithmetic in the sense defined below.

2. ERROR ANALYSIS. Recognition and analysis of the roundoff error inherent in actual computation predates the computer era. Since only a few actual calculations could be performed by hand or mechanically, what is called *forward analysis* [8], in which the propagation of the errors generated at each step was followed into the result, was more or less feasible. The introduction of electronic machines, in which the number of operations progressed from hundreds to thousands to millions, lead to a recognition of the inadequacy of forward error analysis, and the development of other methods of error estimation and control. A bibliography published in 1965 [22] already contained 903 entries relating to the subject. By the time conferences were held on error in digital computation at the Mathematics Research Center in 1964-65 [23], [24], the basic ideas of *backward error analysis* [1], *significance arithmetic* [3], and *interval arithmetic* [15], [16], had already been introduced. However, the inaccurate, erratic arithmetic performed by the floating-point units of most available computers plays havoc with the *analysis* of the error of results, as well as with the accuracy of the results themselves.

Backward error analysis, due to J. H. Wilkinson [27], regards the solution obtained by the computer to be the exact solution of a perturbed version of the original problem, and an estimate of the size of the perturbation is made. The basic formulas [1] are:

$$(2.1) \quad fl(x \circ y) = (x \circ y) (1 + \epsilon), \quad \circ \in \{+, -, \cdot, /\}.$$

In (2.1), $fl(x \circ y)$ denotes the floating-point number actually computed for the exact result $x \circ y$, and ϵ is a machine-dependent number which is a function of the base in which floating-point numbers are represented, the number of digits in their mantissas, and the way in which rounding is done. According to the example given in the Abstract, one has to take $|\epsilon| = 1$ for the UNIVAC 1100, instead of $|\epsilon| = 2^{-27}$ on the basis of the base $b = 2$ and the number of mantissa digits $n = 27$. The error bounds one gets in this case are rigorous, but too large to be useful. The fault does not lie with the theory, but the way the arithmetic unit of this particular machine works. Of course, (2.1) can be used to formulate a *standard*: It is to hold for all defined floating-point operations with $|\epsilon| \leq b^{-n}$ over the entire set of floating-point numbers x, y available on the machine. This criterion, as simple as it is, is not met by many machines on the present market.

Significance arithmetic attempts to model the degradation of accuracy in a calculation by reduction of the number of digits carried as significant. Interval arithmetic, on the other hand, traps the true result of the calculation between two machine numbers, and thus leads to guaranteed lower and upper bounds

for the exact answer. These types of arithmetic carry with them certain aspects of forward analysis. They are, however, useful in connection with the accurate arithmetic to be described below, and interval arithmetic has many other applications [17]. Interval arithmetic for real and complex numbers, vectors, and matrices is included in the theory of computer arithmetic given by Kulisch and Miranker [13], and has been implemented as standard in the language PASCAL-SC [28], about which more will be said later.

The above considerations are concerned mostly with *computational* error, which is produced by the arithmetic unit of the computer, sometimes in a way unknown to the user. This type of error is extremely invidious, like an undetected cancer eating away at the guts of the computation. A more superficial type of error is *conversion* error, which enters into consideration if $b \neq 10$. As in the case of rounding, methods for accurate base conversion are well-understood [30], and it is not too much to require that the machine and the associated input/output software perform conversion as accurately as possible. Once again, this is a standard that is not always met. For example, the following program in standard PASCAL was used to test conversion on a couple of machines.

```

program rw(input, output);
var x: real;
begin while not eof do
(2.2)      begin readln(x); writeln(x)
            end;
end.
```

The results using a VAX 11/780 at the University of Wisconsin-Madison were:

	Input	Output
	1.1	1.1000000 <u>24</u> E+00
	2.1	2.0999999 <u>05</u> E+00
(2.3)	3.1	3.0999999 <u>05</u> E+00
	1.2	1.2000000 <u>48</u> E+00
	2.2	2.2000000 <u>48</u> E+00

Incorrect digits are underlined. Thus, the number of correct digits produced varies between seven and eight out of the ten printed, according to a pattern that may not be immediately apparent. An analysis of the representation of single precision floating-point numbers on the VAX reveals that ten-digit decimal numbers cannot be represented exactly in general. Thus, some of the digits of the ten printed will be fabrications in most cases, which might deceive the uninitiated.

The same program (2.2) was also run on the UNIVAC 1100 system at the University of Wisconsin-Madison, with results which are given in (2.4). Here, fewer digits are printed, but they are all accurate. This is more in the spirit of accurate (and significance) arithmetic. Accuracy is obtained here at the price of sacrifice of two or three additional accurate digits which could have been printed. Thus, inaccurate information is presented in (2.3), and accurate digits are coyly concealed in (2.4). The ideal situation, of course, is for the input/output software to convert as accurately as possible, and if desired, print all

accurate digits of the output, and only those.

	Input	Output
	1.1	1.1000E+00
	2.1	2.1000E+00
(2.4)	3.1	3.1000E+00
	1.2	1.2000E+00
	2.2	2.2000E+00

There is one remedy for conversion error which has been around for a long time, namely, *decimal* arithmetic. (One recalls older machines such as the IBM 650 and the Burroughs 205.) Decimal arithmetic has also been implemented in the microcomputer version of PASCAL-SC [14], [28]: The standard format chosen for floating-point numbers is *twelve* decimal digits for the mantissas, with two decimal digit exponents. The digits in the mantissas are packed as two BCD characters per eight-bit word. In this era of large-scale integration of computer circuits, hardware to handle longer strings of decimal digits should be simple to produce. (On the microcomputer, this is done principally by software.)

Even if the above discussion is unconvincing with regard for the need to keep arithmetic computation and conversion error to a minimum, consider the following situation: Two computations, each costing the same, are made of some quantities of interest. Suppose that it is known that one computation is performed with more accurate arithmetic than the other. Which is preferable, considering that the results may be used to make decisions on which the safety of life and property depend? This is the main point here: Now that the implementation of accurate arithmetic is known to be relatively simple, why not insist upon it?

3. FLOATING-POINT ARITHMETIC. Here, the discussion will be intuitive, using concepts familiar to most users of digital computers. For precise definitions and a rigorously structured argument, see the book by Kulisch and Miranker [13]. The ingredients needed are the set of floating-point numbers, the arithmetic operations, and various roundings. The set of floating-point numbers available on most machines constitutes what will be called a *screen* S ; it is linearly ordered, and contains the identity elements of addition (0) and multiplication (1). S is a subset of the real numbers which contains $-a$ if $a \in S$, and a *maximum* element $s = \max\{a \mid a \in S\}$ (and hence the *minimum* element $-s$). There are, of course, only a finite number $\#S$ of elements of S . Excluding division by zero, the basic *arithmetic operations* $+$, $-$, \cdot , $/$ can be applied to pairs of elements of S to obtain a larger, but still finite, set of real numbers AS consisting of the results. Elements $x \in AS$ such that $|x| > s$ will be said to have *overflowed* S ; in actual practice, the attempt to form such results will lead to an error indication. The remaining elements $y \in AS$ such that $|y| \leq s$ will not, in general, be elements of S ; before computation can proceed, a mapping $y \rightarrow a$, $a \in S$ is required. More generally, we want to consider a class of mappings from the real numbers contained in the interval $[-s, s]$ into S ; such mappings are known as *roundings*.

There are three types of roundings of basic importance for accurate computer arithmetic [13]: There is *upward* rounding Δ , defined by

$$(3.1) \quad \Delta r = \min\{a \mid a \geq r, a \in S\}, \quad r \in [-s, s],$$

and downward rounding ∇ , for which

$$(3.2) \quad \forall r = \max\{a \mid a \leq r, a \in S\}, \quad r \in [-s, s],$$

just as you expected. These are the *monotone* roundings. Another type of rounding, called *antisymmetric* rounding, is denoted by \square ; here $\square r \in S$ and

$$(3.3) \quad \square(-r) = -(\square r).$$

Antisymmetric rounding is probably best illustrated by examples. One way to define $\square r$ is as the *closest* element a of S to r , with a tie-breaking rule satisfying (3.3) if r is equidistant from two elements of S . Thus, if S consists of integers, then $r = 1.6$ would be rounded to $a = 2$, and $r = -1.6$ to $a = -2$, thus satisfying (3.3). This type of rounding has the least possible absolute error, and Yohe [29] calls $\square r$ in this case the Best Possible Answer (BPA). There are other antisymmetric roundings, however. Simple *truncation* rounds $r = 1.6$ to $a = 1$ (downward), and $r = -1.6$ to $a = -1$ (upward), and (3.3) thus also holds in this case, sometimes called rounding *toward zero*. Rounding *away from zero* ($\square(1.6) = 2$, $\square(-1.6) = -2$) is also an antisymmetric rounding, but is less frequently encountered than truncation.

Rounded computer arithmetic, then, consists of arithmetic operations followed by rounding. For accuracy, the antisymmetric rounding of choice is the BPA rounding of the result of the operation to the closest floating-point number. In order to implement this rounding correctly [13], [29] for n -digit mantissas, the accumulator has to be extended to $n + 2$ digits, preceded and followed by one binary digit (bit). The detailed algorithms can be found in [13]. It is also extremely helpful to have a *long* accumulator of $2n + 1$ digits preceded by a bit. The extra expense of providing these accumulators, compared to the usual double-length accumulator for floating-point arithmetic, should be a negligible component of overall computer cost. In addition, the short and long accumulators described also permit the monotone roundings (3.1) and (3.2). Actually, only one monotone rounding need be implemented, the other can be obtained from it by sign changes: $\Delta(a) = -\nabla(-a)$ [13]. However implemented, both monotone roundings are needed for interval arithmetic [13], [17], which is an essential component of accurate computation.

A *minimum* standard for floating-point arithmetic, not met by present commercial units in general, is

$$(3.4) \quad \bigwedge_{\circ \in \{+, -, \cdot, /\}} \bigwedge_{a, b \in S} a \circ b = \square(a \circ b), \quad (a \circ b) \in [-s, s],$$

where \square denotes the BPA rounding to the nearest floating-point number, $a \circ b$ is the *exact real result* of the operation \circ , and $a \circ b$ is the *computed* result actually obtained. In words, (3.4) says that the floating-point number produced by the rounded floating-point arithmetic operation is the closest floating-point number to the exact real result, if defined. Addition of the monotone roundings to (3.4) results in the following STANDARD FOR FLOATING-POINT ARITHMETIC:

$$(3.5) \quad \bigwedge_{\circ \in \{\square, \nabla, \Delta\}} \bigwedge_{\circ \in \{+, -, \cdot, /\}} \bigwedge_{a, b \in S} a \circ b = \circ(a \circ b), \quad (a \circ b) \in [-s, s].$$

Again, $a \oplus b$ denotes the floating-point number produced by the computer, and $a \circ b$ the exact real result of performing the arithmetic operation \circ with the floating-point arguments a and b . Furthermore, the user should be able to select the particular rounding desired. This leads to twelve floating-point arithmetic operations followed by the corresponding rounding:

	$+, -, \cdot, /,$	Best possible answer (BPA) rounding;
(3.6)	$+>, ->, \cdot>, /> ,$	Upward (Δ) rounding;
	$+<, -<, \cdot<, /< ,$	Downward (∇) rounding.

All of these correctly rounded arithmetic operations are standard in PASCAL-SC. The user simply employs the notation on the left side of (3.6) (with \cdot replaced by $*$, as usual; in this paper, $*$ is reserved for another purpose, namely, the scalar product of vectors).

4. HARDWARE vs. SOFTWARE IMPLEMENTATION. Generally speaking, operations which are not provided by the hardware of a computer can be obtained by programming. In general, then, the efficiency of a computation depends on the ratio of hardware to software operations. A *primitive* computer will be defined to be one in which only (accurate) integer arithmetic is available in hardware. Most of the 8-bit and 16-bit microcomputers fall into this category, as well as some 32-bit processors. On primitive machines, all floating-point arithmetic has to be done by software; thus, accurate operations (3.6) can be performed in essentially the same time as the usual inaccurate ones with uncontrolled rounding. The details of implementation of accurate floating-point arithmetic on primitive computers are given in [13], Chapter 6. There is no reason not to use the algorithms in this source, even if only BPA rounding is desired.

Machines with built-in floating-point hardware are found in the minicomputer, standard, and supercomputer ranges. The arithmetic units of these machines will be said to be *useless* for efficient scientific computation unless (3.5) is satisfied. On this basis, almost all available floating-point arithmetic units are junk. The operations in (3.6) which are not provided by the hardware (this often means *all* of them) must be implemented by resort to primitive operations, with a dreadful loss in efficiency. This is well-documented in the case of implementation of interval arithmetic on a number of standard computers [2], [7], [21], [26]. These studies show factors of 50 to 300 in time between wired-in and programmed operations. On a primitive computer (the DEC 10), the ratio between ordinary floating-point arithmetic and interval arithmetic was observed to be 2, which is to be expected, since the calculation of an interval requires computation of its two real endpoints.

One bright note is now the most modern computers have or allow facilities for microprogramming of operations. Thus, it is possible in many cases to obtain the operations in (3.6) with little or no increase in execution time. In fact, Moore [18] reports a ration of 1.8 of microprogrammed interval arithmetic to real arithmetic on the PDP 11/40E, a very ordinary minicomputer. Here, the ratio is less than 2 because of capabilities of the machine for overlapping successive operations.

To summarize, accurate floating-point arithmetic on microcomputers or micro-programmable computers is at present competitive with the uncontrolled, inaccurate kind. If appropriate standards are set by purchasers, then this should lead to machines which are offered with built-in floating-point arithmetic units which are specifically designed for accuracy and implement (3.6) in hardware. Both short and long accumulators should be available.

5. STANDARD FUNCTIONS. In addition to arithmetic, scientific computation of course requires a library of a number of standard functions, and the specifications for accuracy and rounding have to be consistent with those for floating-point arithmetic. If f is such a standard function with domain $D(f) \subset S \cap \mathbb{R}$, then the basic requirement is

$$(5.1) \quad \bigwedge_{\alpha \in \{\square, \nabla, \Delta\}} \bigwedge_{a \in D(f)} \bigcirc f(a) = \alpha f(x),$$

where $\bigcirc f(a)$ is the result actually computed, and $f(a)$ is the *exact* real number defined by the transformation f applied to a . Equation (5.1) embodies the goal of accurate scientific computation: Obtain as the result a floating-point number which is closest to the exact result, or the closest larger or smaller element of S to $f(a)$, as desired. Most software for scientific computation provides a number of standard functions (square root, sine, cosine, exponential, logarithmic, and so on) which appear frequently in scientific formulas, although not necessarily with the accuracy prescribed by (5.1). Once again, since the methods for accurate computation of standard functions are well understood, (5.1) should be adopted as a *standard* requirement for accuracy of functions provided by compilers.

One way to meet the requirement (5.1) in a given precision is to calculate the BPA for $f(a)$ in higher (not necessarily double) precision, and then perform the appropriate rounding to the shorter length. This is the strategy adopted in the PASCAL-SC compiler [28], in which a longer real number with 20 decimal digit mantissa is provided in addition to the standard 12 digit format. Thus, it is also possible to add functions for special computational purposes which also meet the requirement of accuracy (5.1), providing careful error analysis and programming is done.

6. VECTOR AND MATRIX CALCULATIONS. In the area of calculations with vectors and matrices, the theory of computer arithmetic given by Kulisch and Miranker [13] provides a significant advance in accuracy over previous methods. The key to this improvement is the accurate computation of scalar products of vectors, the importance of which was recognized early by Wilkinson [1], [27]. Instead of arising from *ad hoc* considerations, however, the necessity of this accurate scalar product arises from the simple (but deep) concept of a *semimorphism*, defined in [13]. The idea, loosely described, is to represent the algebra of the entities entering into the calculation as accurately as possible by floating-point numbers. In particular, if A is an algebraic system in which the operation \circ is defined, one requires that

$$(6.1) \quad \bigwedge_{a, b \in A} a \square b = \square(a \circ b),$$

where \square is an antisymmetric rounding which preserves the (partial) ordering relation in A . It follows that floating-point arithmetic satisfying (3.1) constitutes an example of (6.1) in which $A = \mathbb{R}$, the set of real numbers. Such computer arithmetic is called *semimorphism-induced* arithmetic.

For $A = VR$, the set of vectors of some given finite dimension over the reals, the operations of addition and subtraction are readily seen to satisfy (6.1) if

done componentwise by arithmetic for which (3.4) holds. Similarly, multiplication of vectors by real numbers can be achieved semimorphically by accurate arithmetic done componentwise. Going on to $A = MR$, the matrices over the real numbers, there is once again no problem with addition, subtraction, or multiplication by real numbers. Vector-matrix, matrix-matrix, or matrix-vector products, however, require calculation of the scalar product

$$(6.2) \quad a \cdot b = \sum_{i=1}^m a_i \cdot b_i$$

of two m -dimensional vectors $a, b \in VR_m$. Simply evaluating the right-hand side

of (6.2) in accurate single precision floating-point arithmetic does not achieve (6.1) with $\circ = *$ in general. Calculation of the sum of products in (6.2) in double precision, as recommended by Wilkinson [1], [27], results in a significant improvement in accuracy, but also fails to achieve a semimorphism in general. What is needed for best results is a way to add the m $2n$ -digit double precision products $a_i \cdot b_i$ without loss of accuracy. This is provided by algorithms due to

Bohlender [5], [26], described in [13]. The resulting semimorphic scalar product is implemented in PASCAL-SC.

There are a number of dramatic illustrations of the accuracy with which tasks of linear algebra can be performed using accurate scalar products. For example, the segment of order 15, $H_{15} = ((i + j - 1)^{-1})$, $i, j = 1, 2, \dots, 15$, of the notorious Hilbert matrix can be inverted on a microcomputer with the result guaranteed to be accurate to 11 of the 12 decimal digits of the mantissas. This is in spite of the fact that the condition number of the matrix is $P(H_{15}) \cong e^{52.5} \approx 10^{23}$, so that, by rule of thumb, a roundoff error of 10^{-12} should correspond to a loss of 11 significant digits, rather than the observed uncertainty of one. (See [1] for an error analysis of matrix inversion based on condition number; a table of test matrices with their condition numbers is given in [19], for example.)

The accuracy cited above is obtained by what is called *horizontal extension* of matrix arithmetic into the floating-point number screen. This concept is illustrated by the appropriate segment of the Kulisch diagram ([13], p. 2):

I	II	III	IV	V
1	R	D	D	S
				+ - . /
				x
2	VR	VD	VD	VS
				+ -
				x
3	MR	MD	MD	MS
				+ - .

Figure 6.1. A segment of the Kulisch diagram.

Interpretation of this diagram can be made on the basis that D is the set of double precision floating-point numbers, and S is the screen of single precision floating-point numbers; VD, VS are the corresponding vectors, and MD, MS matrices over D, S. In each row, the defined arithmetic operations are given in column

V. The \times between rows indicates that multiplications are defined: Vectors and matrices can be multiplied by numbers, and matrices can be multiplied by vectors. It is assumed, of course, that the appropriate dimensions obtain.

The matrix multiplication based on (6.2) satisfies (6.1); it is the direct realization of multiplication in MR in MS. The product built up from evaluation of (6.2) using operations in D or S, shown in column V, results in

$$(6.3) \quad (a,b) = a_1 \square b_1 \oplus a_2 \square b_2 \oplus \dots \oplus a_m \square b_m,$$

where the rounding \square has been indicated for the operations in D or S. This product, often very different in value from $a*b$, is the result of moving downward in the Kulisch diagram, hence, it is said to be obtained by *vertical extension*. The product (6.3) is the one calculated by ordinary matrix/vector software, and leads to the inaccurate results customarily obtained, particularly for poorly conditioned problems.

The basic idea above also comes up in connection with standard functions, and can be stated loosely as follows: One must distinguish between the evaluation of a *formula* for a function by floating-point arithmetic (accurate or not), and the evaluation of the *function* as a floating-point number adjacent to its exact real value. In the latter case, implementation of the operations involved have to be done accurately in order to satisfy the standards (5.1) and (6.1). This can require detailed analysis; the software used and the hardware should assist in attaining the desired goal. Accurate floating-point arithmetic and the features of PASCAL-SC constitute significant advances in this direction.

7. COMPLEX ARITHMETIC. Since complex numbers, vectors, and matrices arise in many scientific computations, accurate arithmetic for these data types is also required. Thus, arithmetic for these types is included in the general theory [3], and implemented in PASCAL-SC according to the standards (3.5) for arithmetic [1] for standard functions [28], and (6.1) for the scalar product (6.2).

8. INTERVAL ARITHMETIC AND INCLUSION ALGORITHMS Interval analysis [17] has applications to many important problems, and is based ultimately on interval arithmetic and interval versions of standard functions. In many cases, intervals can be used to represent guaranteed lower and upper bounds for exact results in scientific calculations, and also to guarantee the *existence* of solutions to problems within those bounds. In order to have computed results in which one can have this kind of confidence, monotone rounding has to be made to S, so that lower endpoints will be rounded downward, and upper endpoints upward. As noted above, the simple provision of the roundings Δ, ∇ in hardware is really all that is basically required for the efficient implementation of interval arithmetic. Even better, the arithmetic unit could be built for direct execution of interval arithmetic.

Interval algorithms which give guaranteed lower and upper bounds for the results of exact real computations will be called *inclusion* algorithms. Such algorithms are known for a wide variety of computational tasks: Solutions of linear and nonlinear systems of equations, inverses of matrices, solutions of ordinary differential and integral equations, and so forth. If the results of an inclusion algorithm are intervals in which the lower and upper endpoints agree to a certain number of places, then the real result is determined to that accuracy. This is used in PASCAL-SC as a significance criterion when inclusion algorithms are employed; only significant digits are printed, so the guaranteed accuracy of

the result is apparent at a glance. Thus, one obtains not only a result, but at the same time the accuracy of the result. This is in stark contrast to what happens in ordinary computation, in which a fixed number of digits are printed in each answer, and one can be completely in the dark as to how many (if any) are accurate.

In many cases, an efficient use of inclusion algorithms can be made by first computing an approximate real result, using accurate arithmetic, and then using this result to construct a small interval to test for inclusion of the exact answer. For an example connected with the use of Newton's method for the solution of nonlinear systems of equations, see [25]. This *postapplication* of interval technique often gives guaranteed existence and error bounds with little additional computation time, as opposed to doing the entire calculation in interval arithmetic. Of course, as advances in hardware result in higher speed for interval arithmetic, this point may not be as important as the possibility, in some cases, that smaller intervals can be obtained by postapplication.

In addition to arithmetic for real intervals, interval vectors, and interval matrices, inclusion algorithms for zeros of polynomials, eigenvalues and eigenvectors, and solutions of nonlinear systems of equations require the corresponding complex arithmetics, since the numbers which will arise in scientific computation involving these problems will be complex in general. Thus, a further requirement for accurate computation is that the hardware and software provide facilities for six additional data types:

- (1) real intervals; (1c) complex intervals;
- (2) real interval vectors; (2c) complex interval vectors;
- (3) real interval matrices; (3c) complex interval matrices;

all with *semimorphism accuracy*.

Including the real and complex numbers, vectors, and matrices, there are thus twelve floating-point data types which should be considered to be standard. (Of course, integer arithmetic is also essential.) All these types are standard in PASCAL-SC, and appear in the complete Kulisch diagram ([13], p. 2), [28]. It would be ideal if the computer hardware were designed according to the general theory of computer arithmetic given in [13], together with the algorithms for its implementation also provided there.

9. A PROGRAMMING LANGUAGE FOR ACCURATE SCIENTIFIC COMPUTATION. Given the operations for accurate floating-point arithmetic with controlled rounding, there remains the problem of software which will enable the user to take advantage of the available accuracy, and the formation of a library of programs of known and guaranteed accuracy to take care of computational tasks frequently encountered. It is possible, of course, to modify an unstructured programming language such as FORTRAN [6] in this way, and the result would undoubtedly be an improvement over what one finds at present. Structured languages, such as PASCAL, however, offer more opportunities because of the ease in which new data types can be introduced. To bring in complex numbers, for example, the declaration

```
(9.1)      type complex = record re,im of real end;
```

does the job in PASCAL. If one is doing a lot of calculations with real polynomials of degree up to some value, then one could declare

```
(9.2)      const deg =      ; (whatever the chosen value)
           type degree = 0..deg;
           polynomial = array[degree] of real;
```

and so on. However, operations between members of new data types have to be done by functions and procedures, as usual. PASCAL-SC [14], however, allows the user to introduce *operators*: The arithmetic symbols $+$, $-$, \cdot , $/$, if appropriate, can be defined for new data types by the user; for example, for fractions, polynomials, and so forth. In addition, operators (unary or binary) can be given names and priorities, and used in expressions for the appropriate data type. Thus, addition of complex numbers is written simply

```
(9.3)      c := a + b;
```

once $+$ is defined for complex numbers by an operator subroutine, standard in PASCAL-SC; (9.3) also applies to intervals, vectors, matrices, polynomials, etc., under the same condition. This simplifies programming considerably in a number of cases. Furthermore, the user has the opportunity of achieving semimorphism accuracy in certain instances. For example, in the multiplication of polynomials $p(x) = p_0 + p_1x + \dots + p_mx^m$ and $q(x) = q_0 + q_1x + \dots + q_nx^n$, one could write

```
(9.4)      r := p*q;
```

in the definition of \cdot for polynomials, one would note that the coefficients

$$(9.5) \quad r_k = \sum_{i=0}^k p_i \cdot q_{k-i}, \quad k = 0, 1, \dots, m+n,$$

of the product polynomial $r(x) = r_0 + r_1x + \dots + r_{m+n}x^{m+n}$ are essentially scalar products, and thus can be computed accurately as described in §6.

Thus, the rôle of software in accurate scientific computation is twofold: It must allow the user to take advantage of whatever accuracy is provided by the hardware, and to achieve accuracy in operations which must be programmed. The version of PASCAL-SC now available for microcomputers meets these conditions; to move in the direction of larger machines will require microprogramming and possible new designs for arithmetic units. However, these can be based on the available general theory of computer arithmetic [13], and so are not beyond the state of the art. It would be particularly helpful to have wired-in interval and complex arithmetic, and accumulators of extended length ($2n + 2$ digits and two bits and $4n + 1$ digits and one bit) to handle the double-length products encountered in the computation of scalar products, in addition to the short and long accumulators defined in §3. Once again, there is precedent for extra-length accumulators, the electro-mechanical IBM 602-A Multiplying Punch provided an accumulator of 120 decimal digits, which could be broken into smaller units for accounting purposes.

Users should set standards, based on (3.5), (5.1), and (6.1) for hardware and software accuracy, to which the industry can respond. An economic pay-off to more accuracy with fewer digits is smaller, faster, and cheaper machines.

REFERENCES

1. E. L. Albasiny. Error in digital solution of linear problems, [23], pp. 131-184, 1965.
2. J. Q. Arnold, F. P. Ford, and R. G. Hetherington. Implementation and evaluation of interval arithmetic software; Report 3: The Honeywell G635 system. Tech. Rept. O-79-1, No. 3, U. S. Army Waterways Experiment Station, Vicksburg, Miss., 1979.
3. R. L. Ashenurst. Techniques for automatic error monitoring and control, [23], pp. 43-59, 1965.
4. G. Bohlender. Genaue Summation von Gleitkommazahlen, Computing, Suppl. 1 (1977), 21-32.
5. G. Bohlender. Genaue Berechnung mehrfacher Summen, Produkte und Wurzeln von Gleitkommazahlen und allgemeine Arithmetik in Höheren Programmiersprachen, Dissertation, University of Karlsruhe, Germany, 1978.
6. G. Bohlender, E. Kaucher, R. Klatte, U. Kulisch, W. L. Miranker, Ch. Ullrich, and J. Wolff von Gudenberg. FORTRAN for contemporary numerical computation, Report RC 8348, IBM Thomas J. Watson Research Center, White Plains, New York, 1980.
7. D. A. Cohn, J. B. Potter, and M. Ginsberg. Implementation and evaluation of interval arithmetic software; Report 5: The CDC CYBER 70 system. Tech. Rept. O-79-1, No. 5, U. S. Army Waterways Experiment Station, Vicksburg, Miss., 1979.
8. A. S. Householder. The generation of error in digital computation, Rept. No. 1983, Oak Ridge National Laboratory, Oak Ridge, Tenn., 1955.
9. U. Kulisch. An axiomatic approach to rounded computations, MRC Tech. Summary Rept. No. 1020, University of Wisconsin-Madison, 1969; Numer. Math. 19 (1979), 1-17.
10. U. Kulisch. On the concept of a screen, MRC Tech. Summary Rept. No. 1084, University of Wisconsin-Madison, 1970; Z. Angew. Math. Mech. 53 (1973), 115-119.
11. U. Kulisch. Rounding invariant structures, MRC Tech. Summary Rept. No. 1103, University of Wisconsin-Madison, 1970.
12. U. Kulisch. Interval arithmetic over completely ordered ringoids, MRC Tech. Summary Rept. No. 1105, University of Wisconsin-Madison, 1970.
13. U. Kulisch and W. L. Miranker. Computer Arithmetic in Theory and Practice, Academic Press, New York, 1981.
14. U. Kulisch and H.-W. Wippermann. PASCAL-SC: Pascal for Scientific Computation, Institute for Applied Mathematics, University of Karlsruhe, Germany, 1980.
15. R. E. Moore. The automatic analysis and control of error in digital computation based on the use of interval numbers, [23], pp. 61-130, 1965.
16. R. E. Moore. Automatic local coordinate transformations to reduce the growth of error bounds in interval computation of solutions of ordinary differential equations, [24], pp. 103-140, 1965.
17. R. E. Moore. Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics 2, Soc. for Ind. and Appl. Math., Philadelphia, 1979.

18. R. E. Moore. New results on nonlinear systems, [20], pp. 165-180, 1980.
19. M. Newman and J. Todd. The evaluation of matrix inversion programs, J. Soc. Indust. Appl. Math. 6 (1958), 466-476.
20. K. Nickel (Ed.). Interval Mathematics 1980, Academic Press, New York, 1980.
21. S. Podlaska-Lando, E. K. Reuter, and B. D. Schriver. Implementation and evaluation of interval arithmetic software; Report 2: The Honeywell MULTICS system, Tech. Rept. O-79-1, No. 2, U. S. Army Waterways Experiment Station, Vicksburg, Miss., 1979.
22. L. B. Rall. Bibliography on error in digital computation, [23], pp. 207-320, 1965.
23. L. B. Rall (Ed.). Error in Digital Computation, Vol. 1, Wiley, New York, 1965.
24. L. B. Rall (Ed.). Error in Digital Computation, Vol. 2, Wiley, New York, 1965.
25. L. B. Rall. A comparison of the existence theorems of Kantorovich and Moore, SIAM J. Numer. Anal. 17 (1980), 148-161.
26. R. G. Ward. Implementation and evaluation of interval arithmetic software; Report 4: The IBM 370, DEC 10, and DEC PDP 11/70 systems, Tech. Rept. O-79-1, No. 4, U. S. Army Waterways Experiment Station, Vicksburg, Miss., 1979.
27. J. H. Wilkinson. Rounding Errors in Algebraic Processes, Prentice-Hall, Englewood Cliffs, N.J., 1963.
28. J. Wolff von Gudenberg. Gesamte Arithmetik des PASCAL-SC Rechners: Benutzerhandbuch, Institute for Applied Mathematics, University of Karlsruhe, Germany, 1981.
29. J. M. Yohe. Best possible floating point arithmetic, MRC Tech. Summary Rept. No. 1054, University of Wisconsin-Madison, 1970.
30. J. M. Yohe. Accurate conversion between number bases, MRC Tech. Summary Rept. No. 1109, University of Wisconsin-Madison, 1970.
31. J. M. Yohe. Roundings in floating-point arithmetic, IEEE Trans. Computers C-22 (1973), 577-586.

Mathematical Software and Mathematical Software Libraries

Alfred H. Morris, Jr.

Naval Surface Weapons Center
Dahlgren, Virginia 22448

ABSTRACT. A brief summary of the evolution of general-purpose mathematical software development is given. The NSWC library is then introduced, and sources of high-quality numerical mathematical software are provided.

1. Background

Attitudes concerning the development of numerical mathematical software libraries have changed considerably in the last 20 years. In the early and middle 1960's most organizations that had a computer found it convenient to have a library of routines which everyone could use. Normally the library was just a repository for commonly used software. If a routine was found to be particularly useful, then more often than not the routine was blindly dumped into the library. As a result, most libraries contained a mixture of good, mediocre, and unbelievably bad routines.

In the latter 1960's and early 1970's, because of the increased cost in the production and maintenance of software, the increased complexity of the problems being considered, and the general unreliability of many of the existing codes, the relaxed attitudes concerning the formation of libraries began to change. By now it was clear that any laboratory which employed computers for a variety of scientific applications should contain a library of accurate, efficient, general-purpose subroutines. It was also clear that the formation of such a library was not a simple task. For example, the development of high-quality software frequently required technical expertise that was not available in-house. Also, it frequently required the development of new mathematical theory, which could be an arduous and expensive process.

In 1971 the NATS (National Activity to Test Software) project began. NATS was a joint effort by the Atomic Energy Commission and the National Science Foundation to examine the problems, costs, and resources involved in the production, certification, dissemination, and maintenance of high-quality mathematical software. The project

was centered at Argonne National Laboratory, and involved the collaborative effort of individuals at some two dozen university and research laboratories. The initial product of NATS was EISPACK, a comprehensive package of FORTRAN subroutines for eigenvalue/eigenvector computation. Exceedingly high quality control was maintained in the development of EISPACK. This package has had considerable impact. It is extensively used, and it helped to establish the minimal software engineering standards that currently exist.

In 1970 development of the IMSL and NAG libraries began. The IMSL library was a commercial venture by the International Mathematical and Statistical Libraries corporation. To gain acceptance, emphasis was placed on developing a quality library that was comprehensive in both mathematics and statistics, thereby providing greater capability than most laboratory libraries possessed. The NAG (Numerical Algorithms Group) project began as a joint effort by British universities and government research laboratories to produce a comprehensive numerical library. The effort is now established as a non-profit organization. Part of NAG's income is derived from renting its library, which places it in direct competition with IMSL. The IMSL and NAG libraries are good libraries. These and other commercial libraries have had a considerable impact on the activities of many organizations, providing a broad capability for a variety of computers at an economical price.

Since the late 1960's Sandia Laboratories and several other organizations have also begun a methodical development of high-quality numerical mathematical libraries. The purpose for the formation of these libraries was not to lease or sell software, but to provide quality software for in-house use and for general use by other organizations. The development of these libraries represents a significant research and development investment. In many cases, a laboratory has leased a commercial library while developing its own library. When this occurs, normally the two libraries (the leased library and the in-house library) are kept separate from one another. They tend to provide complementary rather than duplicate capability. Currently, all libraries are deficient to some degree. Deficiencies cannot be avoided, since there are possibly more numerical mathematical questions that have not yet been answered than have been answered.

2. Formation of the NSWC Library

In 1976 formation of the NSWC library of numerical mathematics subroutines began. The objective was to form a high-quality library of general-purpose subroutines that would provide a basic capability in a variety of mathematical activities. The routines were to be written in FORTRAN. Even though the routines were intended for use on the CDC 6000 series computers, every attempt was to be made to ensure their transportability.

The routines in the NSWC library are selected from a variety of sources. Obtaining suitable sources is a difficult task. It is

frequently made more difficult by the requirement that the library routines be nonproprietary. Proprietary restrictions can severely inhibit the use of software, and the subsequent research and development that the software can generate.

All routines are subject to evaluation and possible modification before being accepted for the NSWC library. Primary considerations include the reliability and transportability of the routine, its efficiency and ease of use, and the generality of the routine. In regard to reliability, the major concerns are accuracy, the mathematical stability of the algorithm being employed, and the routine's robustness. The routine is tested, portions of its code are examined, and an assessment is made of the utility and overall performance of the routine. All routines in the library are periodically reviewed for possible improvement. When better routines are obtained then the older routines are eliminated.

In regard to transportability, it is clear that machine dependent constants and precision dependent algorithms cannot be avoided. However, machine dependent code is not permitted. It is assumed that the FORTRAN compiler does not alter arithmetic expressions, and that the floating point arithmetic being employed satisfies criteria such as:

- 1) Additive symmetry; i.e., $-x$ is representable as a floating point number if x is a floating point number.
- 2) All small integers are represented exactly in the floating point arithmetic.

To date, no criteria have been formulated for avoiding the problems that can arise when these or similar conditions are violated. Generally, the policy is to accept code only if it is transportable; i.e., only if its transference to a different computer environment requires changes which are capable of being implemented by a pre-processor. Sufficient documentation must, of course, be supplied to clarify all conversion ambiguities.

The ease of use criterion for the NSWC library software is of considerable importance. The main purpose of the library is to provide a service to as broad an audience as possible. Thus, it is important that duplicate abilities be kept to a minimum, and that the routines be as simple to use and as comprehensive in scope as is practical. To meet these specifications, many specialized sub-routines are incorporated into the library at a subordinate level, being referenced by simple-to-use driver routines. The driver routines are fully documented in the NSWC library reference manual [22], but the supportive routines are only referenced. The policy of referencing supportive code, thereby inhibiting its use except by the specialist, makes it possible to replace the code with minimal impact to the laboratory. Also, it significantly simplifies the situation for the novice (and most users at NSWC), thereby promoting greater and better use of the software than otherwise could be expected.

Development of software that satisfies the ease of use criterion can be characterized as a packaging problem, the objective being to package mathematical theory and formulae into comprehensive, simple-to-use subroutines. It would appear that the importance of this objective would be self-evident, but this is not always the case. It occasionally occurs that a researcher will design a beautiful algorithm. He will exercise great care in the development of subroutines which compute separate portions of the algorithm, and then he will link the subroutines together by a poorly conceived driver routine that is difficult or almost impossible to use. When this occurs, the evaluator is frequently forced to either reject the software, or to completely repackage the software.

In the packaging of software, extreme caution should be taken not to unnecessarily restrict the scope and versatility of the code. Currently the only requirement for the library software that has a direct bearing upon this issue involves the use of I/O. No print statements are permitted for reporting errors. If error detection is performed in a routine, then it is required that the call line of the routine contain a parameter which can be used for reporting the error. The use of such a parameter permits the user to control the sequence of events that occur when an error arises.

The evaluation of software for the NSWC library includes examination of the algorithm and portions of the code, and testing the software. The testing serves many purposes, including determination of the accuracy and efficiency of the software, checking for defects in the code, and searching for regions of instability. Because of the theoretical complexity of many of the mathematical activities being computerized, only infrequently can the testing be complete. Normally the testing will be highly selective, being used to locate and examine weaknesses in the algorithm and code.

3. The NSWC Library Software

The current edition of the NSWC library contains 343 routines, 211 of which are documented in the library reference manual [22]. The remaining routines (the supportive routines) are referenced when it is appropriate to do so. In this section a brief outline of the major library routines is provided. It will be noted that certain sections of the library (e.g., the Special Function section) are unusually comprehensive in scope, whereas other sections (e.g., the Optimization section) are still in their infancy. Approximately 40% of the routines were developed at NSWC, the remainder originating from a variety of sources.

Special Functions

Real and complex routines are provided for computing the error and Fresnel integral functions, the exponential integral function, the gamma and digamma functions, and the ordinary and modified Bessel functions [2,3,15,16]. Also real routines are available for

computing the incomplete and inverse incomplete gamma ratio functions, the complete and incomplete elliptic integrals of the first, second, and third kinds [9,18], the Weierstrass elliptic function for the equianharmonic and lemniscatic cases [11], and the circular and elliptical coverage functions.

Solutions of Nonlinear Equations

A modified form of the ZEROIN routine by Forsythe, Malcolm, and Moler [12] is available for finding zeros of functions of a single variable. The code is an adaptation of the ALGOL 60 procedure ZERO by Brent [5]. Also available is a routine by Jenkins [17] for finding the roots of polynomials, and a MINPACK-1 routine [21] for solving systems of nonlinear equations.

Vectors

BLAS routines [20] for performing elementary vector operations (such as scaling and adding) are provided.

Matrices

Included are routines for performing elementary matrix operations, both in the standard storage format and in sparse form. The routines for the in-place transposition of matrices in the standard format are due to Brenner [4]. Also included are LINPACK routines [10] for inversion and singular value decomposition of matrices in the standard storage format, and code by Sherman [26] for the solution of sparse systems of linear equations.

Eigenvalues and Eigenvectors

The EISPACK routines [28] for computing eigenvalues and eigenvectors appear in a supportive capacity.

Least Squares Solutions of Linear Equations

Included are routines for finding least squares solutions for systems of linear equations with linear equality and inequality constraints. Iterative improvement is performed in several of the routines. The codes were written by Tsao and Nikolai [29], Lawson, Hanson, and Haskell [13,19], and Wampler [30].

Optimization

A MINPACK-1 routine [21] is provided for computing the unconstrained minimum of the sum of squares of nonlinear functions. Also included are routines for solving linear programming problems. In the linear programming routines, the inverse of the basis matrix is computed and stored in core.

Transforms

The Fast Fourier transform code developed by Richard Singleton [27] is provided.

Approximations of Functions

Rational minimax approximation using the Remes-type algorithm designed by Cody, Fraser, and Hart [8] is available. Also available is a modified version of the code by Rice [24] for the L_p approximation of functions.

Curve Fitting

Linear, Lagrange, Hermite, and cubic spline interpolation routines are provided. Also available are least squares polynomial approximation routines, and the Cline routines [7] for spline under tension interpolation.

Surface Fitting

Bi-spline under tension interpolation routines are provided. Also Akima's routines [1] are available for surface interpolation for arbitrarily positioned data points.

Numerical Integration

Included are a modified version of Patterson's routine QSUBA [23] and an adaptive Romberg/Newton Cotes procedure for computing definite integrals.

Ordinary Differential Equations

Represented is the work of L. Shampine [12,14,25] for solving nonstiff initial value problems, and a modified form of the routine EPISODE by Byrne and Hindmarsh [6] for solving stiff initial value problems.

References

- [1] Akima, Hiroshi, "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points," ACM Trans. Math Software 4 (1978), pp.148-159.
- [2] Amos, D.E., Daniel, S.L., and Weston, M.K., CDC 6600 Subroutines for Bessel Functions $J_\nu(x)$, $x \geq 0$, $\nu \geq 0$ and Airy Functions $A_1(x)$, $A_2(x)$, $-\infty < x < \infty$, Report SAND 75-0147, Sandia Laboratories, Albuquerque, New Mexico, 1975.
- [3] Amos, D.E. and Daniel, S.L., A CDC 6600 Subroutine for Bessel Functions $I_\nu(x)$, $\nu \geq 0$, $x \geq 0$, Report SAND 75-0152, Sandia Laboratories, Albuquerque, New Mexico, 1975.
- [4] Brenner, N., "Algorithm 467. Matrix Transposition in Place," Comm. ACM 16 (1973), pp.692-694.
- [5] Brent, R., Algorithms for Minimization without Derivatives, Prentice-Hall, 1973.
- [6] Byrne, G.D. and Hindmarsh, A.C., "A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations," ACM Trans. Math Software 1 (1975), pp.71-96.
- [7] Cline, A.K., "Scalar and Planar Valued Curve Fitting using Splines under Tension," Comm. ACM 17 (1974), pp.218-220.
- [8] Cody, W.J., Fraser, W., and Hart, J.F., "Rational Chebychev Approximation using Linear Equations," Numerische Mathematik 12 (1968), pp.242-251.
- [9] DiDonato, A.R. and Hershey, A.V., "New Formulas for Computing Incomplete Elliptic Integrals of the First and Second Kind," J. ACM 6 (1959), pp.515-526.
- [10] Dongarra, J.J., Bunch, J.R., Moler, C.B., and Stewart, G.W., LINPACK User's Guide, Society for Industrial and Applied Mathematics, Philadelphia, 1979.
- [11] Eckhardt, Ulrich, "Algorithm 549, Weierstrass' Elliptic Functions," ACM Trans. Math Software 4 (1980), pp.112-120.
- [12] Forsythe, G.E., Malcolm, M.A., and Moler, C.B., Computer Methods for Mathematical Computations, Prentice-Hall, 1977.
- [13] Haskell, K.H. and Hanson, R.J., Selected Algorithms for the Linearly Constrained Least Squares Problem - A User's Guide, Report SAND 78-1290, Sandia Laboratories, Albuquerque, New Mexico, 1979.
- [14] Haskell, K.H. and Jones, R.E., Brief Instructions for Using the Sandia Mathematical Subroutine Library, Report SAND 77-1441, Sandia Laboratories, Albuquerque, New Mexico, 1978.

- [15] Hershey, A.V., Approximation of Functions by Sets of Poles, Technical Report TR-2564, Naval Weapons Laboratory, Dahlgren, Virginia, 1971.
- [16] -----, Computation of Special Functions, Technical Report TR-3788, Naval Surface Weapons Center, Dahlgren, Virginia, 1978.
- [17] Jenkins, M.A., "Zeros of a Real Polynomial," ACM Trans. Math Software 1 (1975), pp.178-189.
- [18] Koos, R.L., Numerical Evaluation of the Elliptic Integral of the Third Kind, Technical Note TN-K/22-69, Naval Weapons Laboratory, Dahlgren, Virginia, 1969.
- [19] Lawson, C.L. and Hanson, R.J., Solving Least Squares Problems, Prentice-Hall, 1974.
- [20] Lawson, C.L., Hanson, R.J., Kincaid, D.R., and Krogh, F.T., Basic Linear Algebra Subprograms for FORTRAN Usage, Report SAND 77-0898, Sandia Laboratories, Albuquerque, New Mexico, 1977.
- [21] More, J.J., Garbow, B.S., Hillstom, K.E., User Guide for MINPACK-1, Report ANL-80-74, Argonne National Laboratory, Argonne, Illinois, 1980.
- [22] Morris, A.H., NSWC/DL Library of Mathematics Subroutines, Technical Report NSWC TR 81-410, Naval Surface Weapons Center, Dahlgren, Virginia, 1981.
- [23] Patterson, T.N.L., "Algorithm for Automatic Numerical Integration Over a Finite Interval," Comm. ACM 16 (1973), pp.694-699.
- [24] Rice, J.R., "Algorithm 525. ADAPT, Adaptive Smooth Curve Fitting," ACM Trans. Math Software 4 (1978), pp.82-94.
- [25] Shampine, L.F. and Gordon, M.K., Computer Solution of Ordinary Differential Equations, W.H. Freeman and Company, 1975.
- [26] Sherman, Andrew, "Algorithms for Sparse Gaussian Elimination with Partial Pivoting," ACM Trans. Math Software 4 (1978), pp.330-338.
- [27] Singleton, R.C., "An Algorithm for Computing the Mixed Radix Fast Fourier Transform," IEEE Trans. Audio and Electroacoustics, vol. AU-17 (1969), pp.93-103.
- [28] Smith, B.T., Boyle, J.M., et al., Matrix Eigensystem Routines - EISPACK Guide (Second Edition), Springer-Verlag, 1976.
- [29] Tsao, N.K. and Nikolai, P.J., Procedures using Orthogonal Transformations for Linear Least Squares Problems, Report ARL TR 74-0124, Aerospace Research Laboratories, 1974.
- [30] Wampler, Roy, "Solutions to Weighted Least Squares Problems by Modified Gram-Schmidt with Iterative Refinement," ACM Trans. Math Software 5 (1979), pp.457-465.

ADI PROCEDURES FOR SOLVING THE SHALLOW-WATER EQUATIONS
IN TRANSFORMED COORDINATES

H. L. Butler

U.S. Army Engineer Waterways Experiment Station
P.O. Box 631, Vicksburg, Miss. 39180

Y. P. Sheng

Aeronautical Research Associates of Princeton, Inc.
P.O. Box 2229, Princeton, New Jersey, 08540

ABSTRACT. In order to study the dynamic response of coastal water to tides (astronomical or storm included), tsunamis, and/or meteorological forcing, a two- or three-dimensional free-surface time-dependent model is often desired. However, most such models require an exceedingly small time step (associated with the propagation of gravity waves over the distance of a horizontal grid spacing), and hence their applications are limited. For model efficiency alternating direction implicit (ADI) procedures are used to solve the vertically-integrated equations of momentum and continuity for the two-dimensional model as well as for the external mode of the three-dimensional model. A major advantage of the subject models is the capability of applying a horizontal coordinate transformation in the form of a piecewise exponential stretch. This procedure results in the application of a smoothly varying grid to a given study region permitting simulation of a complex landscape by locally increasing grid resolution and/or aligning grid coordinates along physical boundaries. Reference is drawn to various applications of the two-dimensional model.

1. INTRODUCTION. Various mathematical models have been developed to investigate the hydrodynamic processes of large bodies of water including the design, operation, and maintenance of various coastal projects. This paper discusses the mathematical development of a two-dimensional finite difference model (Butler, 1980) as well as treatment of the external mode of a three-dimensional model (Sheng, 1981). Both the two- and the three-dimensional models have been, and are being, applied to a variety of Corps of Engineers studies.

A two-dimensional model known as the Waterways Experiment Station (WES) Implicit Flooding Model (WIFM), was first devised for application in simulating tidal hydrodynamics of Great Egg Harbor and Corson Inlets, New Jersey (Butler, 1978a). Program WIFM originally employed an implicit solution scheme similar to that developed by Leendertse (1970) and has been applied in numerous studies where tidal, storm surge, and tsunami inundation phenomena were simulated. Basic features of the model include flood modeling of low-lying terrain, treatment of subgrid barrier effects, and a variable grid option. Included in the model are actual bathymetry and topography, time and spatially variable bottom roughness, inertial forces due to advective and Coriolis acceleration, rainfall, and spatial and time-dependent wind fields. Horizontal diffusion terms in the momentum equations are optionally present and can be used, if desired, for aiding stability of the numerical solution.

In a three-dimensional hydrodynamic model of lake currents, Sheng et al. (1978) separated the computation of three-dimensional velocities (internal mode) which are governed by slower internal dynamics from the computations of water level and mass fluxes (external mode) which are governed by fast surface waves. This mode-splitting technique resulted in significant improvement in numerical efficiency over the earlier three-dimensional model of Sheng (1975). Recently, in developing a generalized three-dimensional model for coastal applications, Sheng (1981) adopted a two time level ADI scheme for computation of the external mode (water level and mass fluxes). This ADI scheme is discussed here. A complete description of the three-dimensional model is found in Sheng and Butler (1982) in these same proceedings.

2. GOVERNING EQUATIONS. The basic equations used in modeling hydrodynamics of inland and coastal waters are derived from the classical Navier-Stokes equations in a Cartesian coordinate system (Figure 1). By assuming (a) the pressure varies hydrostatically in the vertical direction; (b) density variations are negligible except in the buoyancy term; and (c) eddy coefficients are used to account for turbulent diffusion effects, the equations of conservation of mass and momentum are:

$$u_x + v_y + w_z = 0 \quad (1)$$

$$u_t = -\frac{1}{\rho} p_x - ((u^2)_x + (uv)_y + (uw)_z) + fv + (A_H u_x)_x + (A_H u_y)_y + (A_V u_z)_z \quad (2)$$

$$v_t = -\frac{1}{\rho} p_y - ((uv)_x + (v^2)_y + (vw)_z) -fu + (A_H v_x)_x + (A_H v_y)_y + (A_V v_z)_z \quad (3)$$

$$p_z = -\rho g \quad (4)$$

where u , v , and w are the three-dimensional velocities in the x , y , and z directions; t is time; f is the Coriolis parameter; g is the acceleration of gravity; p is the pressure; ρ is the fluid density; A_H and A_V are the horizontal and vertical eddy coefficients. Omitted for brevity are equations for conservation of salinity and temperature, an equation of state, and appropriate boundary conditions.

If the additional assumption of fluid homogeneity is made and a depth-averaging process applied one can derive the usual two-dimensional form or external mode of the governing equations, namely:

$$\eta_t + U_x + V_y = 0 \quad (5)$$

$$U_t = -gd\eta_x - ((U^2)_x + (UV)_y) + fV + E_H (U_{xx} + U_{yy}) +$$

$$\tau_{xs} - \tau_{xb} = -gd\eta_x + M_x \quad (6)$$

$$V_t = -gd\eta_y - ((UV)_x + (V^2)_y) - fU + E_H (V_{xx} + V_{yy})$$

$$\tau_{ys} - \tau_{yb} = -gd\eta_y + M_y \quad (7)$$

where U and V are the vertically-integrated mass fluxes; η is the water surface elevation; d is the local water depth; τ_{xs} and τ_{ys} are shear stresses at the free surface; τ_{xb} and τ_{yb} are bottom shear stresses; and E_H is a horizontal eddy coefficient.

The discussions that follow will concentrate on solving the vertically-integrated equations (5-7). These equations, along with appropriate boundary conditions, completely define the WIFM model, but only the external mode of the three-dimensional model. In the three-dimensional model the internal mode of the flow as described by the three-dimensional velocities (u, v, w) is governed by the equations for the perturbation velocities defined as $u' = u - U/d$ and $v' = v - V/d$. These equations and their computational algorithms are discussed in Sheng and Butler (1982).

3. ADI FINITE DIFFERENCE SCHEMES. The differential equations (5-7) are to be approximated by difference equations. To illustrate how various implicit schemes can be derived consider the simplified linearized matrix equation for these equations:

$$W_t + A W_x + B W_y = 0 \quad (8)$$

where

$$W = \begin{pmatrix} \eta \\ U \\ V \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 \\ gd & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \quad B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ gd & 0 & 0 \end{pmatrix}$$

It will be convenient to introduce a general class of finite difference schemes as:

$$\frac{1}{\Delta t} (W^{n+1} - W^n) + \left(\frac{A}{\Delta x} \delta_x + \frac{B}{\Delta y} \delta_y \right) (\theta W^{n+1} + (1 - \theta) W^n) = 0 \quad (9)$$

where δ_x and δ_y are central spatial difference operators and θ is a weighting factor, $0 \leq \theta \leq 1$. If $\theta = 0$ the above difference equation would yield the standard two-level explicit scheme. If $\theta > 0$ the resulting schemes are implicit in that more information is required than this one matrix equation provides to solve for the values of W (resulting from the spatial difference operators times θW^{n+1}) at time level, $n+1$. A value of $\theta = 1/2$ yields the well-known Crank-Nicholson scheme while $\theta = 1$ yields a fully implicit scheme.

For ease of analysis equation (9) with $\theta = 1/2$ can be rewritten as:

$$(1 + \lambda_x + \lambda_y) W^{n+1} = (1 - \lambda_x - \lambda_y) W^n \quad (10)$$

where

$$\lambda_x = \frac{1}{2} \frac{\Delta t}{\Delta x} A \delta_x \text{ and } \lambda_y = \frac{1}{2} \frac{\Delta t}{\Delta y} B \delta_y$$

By adding the quantity $\lambda_x \lambda_y (W^{n+1} - W^n)$ to permit factorization, the following relation is obtained:

$$(1 + \lambda_x) (1 + \lambda_y) W^{n+1} = (1 - \lambda_x) (1 - \lambda_y) W^n \quad (11)$$

It can be shown that the addition of the extra term is equivalent to the addition of the truncation error

$$\frac{\Delta t^2}{4} AB \frac{\partial^3 W}{\partial t \partial x \partial y}$$

Thus, the factorized finite difference equation (11) is still a second-order approximation to the differential equation (8). The advantage of using equation (11) lies in the fact that the solution of the factorized form can be split into two separate one-dimensional operations (the ADI approach).

By introducing an intermediate value, W^* , equation (11) can be split into various two-step operations. The most widely used scheme has the following structure:

$$(1 + \lambda_x) W^* = (1 - \lambda_y) W^n \quad (12)$$

$$(1 + \lambda_y) W^{n+1} = (1 - \lambda_x) W^* \quad (13)$$

When the intermediate level, W^* , is eliminated between equations (12 and 13) the factorized form (equation 11) is recovered. The solution of the two-step operation is apparent. A double-sweep solution technique is used to solve equation (12) for W^* assuming values for W^n are known. The full solution, W^{n+1} , is obtained from equation (13), again using a double-sweep procedure.

Other splitting methods are presented in a later section on the discussion of the subject models.

4. COORDINATE TRANSFORMATIONS. A major advantage of the subject codes is the capability of applying a smoothly varying grid to a given study region permitting simulation of a complex landscape by locally increasing grid resolution and/or aligning coordinates along physical boundaries. For each direction, a piecewise reversible transformation, which takes the form:

$$\begin{aligned}x &= a_x + b_x \alpha^{c_x} \\y &= a_y + b_y \gamma^{c_y}\end{aligned}\tag{14}$$

where coefficients $a_x, a_y, b_x, b_y, c_x, c_y$ are arbitrary and to be determined, is independently used to map prototype or real space into computational space. Variables α and γ are directions in computational space. This procedure treats the computational domain as consisting of a number of regions for which different sets of equations (as in (14) above) apply. The mapping coefficients are determined from an iterative procedure by matching the coordinates and stretching rates, $dx/d\alpha$ or $dy/d\gamma$, at boundaries of adjacent regions (as illustrated in Figure 2).

The stretching does not introduce any additional terms to the equations, but changes the horizontal gradient terms. The resulting equations are:

$$\eta_t + \frac{1}{\mu_x} U_x + \frac{1}{\mu_y} V_y = 0\tag{15}$$

$$U_t = - \frac{gd}{\mu_x} \eta_x + M_x^\mu\tag{16}$$

$$V_t = - \frac{gd}{\mu_y} \eta_y + M_y^\mu\tag{17}$$

where $\mu_x = dx/d\alpha$ and $\mu_y = dy/d\gamma$ and M_x^μ, M_y^μ represent transformation effects on remaining components of the momentum equations. Examples of the stretching procedure are presented in a later section on model application.

The bottom topography in coastal waters, estuaries, and lakes often exhibit significant variation in the horizontal directions. When computing three-dimensional currents, in order to maintain the same order of numerical accuracy in the vertical direction, the (x, y, z) coordinate system (or (α, γ, z) system) is stretched in the vertical direction into a new (x, y, σ) system (or (α, γ, σ) system), such that an equal number of grid points exist in the shallow coastal and the deep offshore areas. The transformation takes the form $\sigma = z/h(x, y)$ where $h(x, y)$ is the local still water depth of the model basin. The equations resulting from this transformation are presented by Sheng, et al. (1978). Notice that for the external mode of the three-dimensional model, as a result of stretching the vertical coordinate, the M_x and M_y terms in equations (16 and 17) will contain a few extra terms than in the two-dimensional model.

5. MODEL ALGORITHMS.

5.1. WIFM. The scheme used in program WIFM is a "leap-frog", three time level scheme which can be derived from the differential equation:

$$\hat{W}_t + A\hat{W}_x + B\hat{W}_y + \hat{M} = 0 \quad (18)$$

where the component variables of \hat{W} are expressed in velocity form and

$$\hat{W} = \begin{pmatrix} \eta \\ u \\ v \end{pmatrix}, \quad A = \begin{pmatrix} 0 & d/\mu_x & 0 \\ g/\mu_x & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 & d/\mu_x \\ 0 & 0 & 0 \\ g/\mu_x & 0 & 0 \end{pmatrix}, \quad \hat{M} = \begin{pmatrix} 0 \\ M_x \\ M_y \end{pmatrix},$$

with \hat{M} containing the velocity form of the nonlinear terms in the governing equations. The approximating difference equation is written as:

$$(1 + 2\lambda_x + 2\lambda_y) \hat{W}^{n+1} = (1 - 2\lambda_x - 2\lambda_y) \hat{W}^{n-1} - 2\Delta t \hat{M}^n \quad (19)$$

By factorizing equation (19) and introducing an intermediate level, \hat{W}^* the structure for WIFM's solution algorithm can be written as:

$$(1 + 2\lambda_x) \hat{W}^* = (1 - 2\lambda_x - 4\lambda_y) \hat{W}^{n-1} - 2\Delta t \hat{M}^n \quad (20)$$

$$(1 + 2\lambda_y) \hat{W}^{n+1} = \hat{W}^* + 2\lambda_y \hat{W}^{n-1} \quad (21)$$

The double-sweep solution technique is used to solve each operational step. A functional representation of each sweep can be expressed as:

x-sweep

$$\eta^* = F_1 [\eta^n, \eta^{n-1}, u^*, u^n, u^{n-1}, v^{n-1}] \quad (22)$$

$$u^* = F_2 [\eta^*, \eta^n, \eta^{n-1}, u^*, u^n, u^{n-1}, v^n, v^{n-1}] \quad (23)$$

$$v^* = F_3 [\eta^n, \eta^{n-1}, u^n, u^{n-1}, v^n, v^{n-1}] \quad (24)$$

y-sweep

$$\eta^{n+1} = G_1 [\eta^*, v^{n+1}, v^{n-1}] \quad (25)$$

$$u^{n+1} = G_2 [u^*] = u^* \quad (26)$$

$$v^{n+1} = G_3 [\eta^{n+1}, \eta^n, \eta^{n-1}, u^n, u^{n-1}, v^{n+1}, v^*, v^n, v^{n-1}] \quad (27)$$

Noting that v^* is an explicit expression, a substitution of F_3 (equation (24)) into G_3 (equation (27)) is made. Thus, each sweep consists in solving a one-dimensional problem involving η^* and u^* in the x-sweep and η^{n+1} and v^{n+1} in the y-sweep.

5.2. External Mode of 3-D Model. Sheng (1981) implemented a two time level fully implicit ADI scheme in computing the external mode of the three-dimensional model:

$$(1 + 2\lambda_x) W^* = (1 - 2\lambda_y) W^n + \Delta t M^n \quad (28)$$

$$(1 + 2\lambda_y) W^{n+1} = W^* + 2\lambda_y W^n \quad (29)$$

A functional representation of each sweep can be expressed as:

x-sweep

$$\eta^* = F_1 (\eta^n, U^*, U^n, v^n) \quad (30)$$

$$U^* = F_2 (\eta^*, \eta^n, U^n, v^n) \quad (31)$$

$$V^* = F_3 (\eta^n, U^n, v^n) \quad (32)$$

y-sweep

$$\eta^{n+1} = G_1 (\eta^*, v^{n+1}, v^n) \quad (33)$$

$$U^{n+1} = U^* \quad (34)$$

$$v^{n+1} = G_3 (\eta^{n+1}, \eta^n, v^*) \quad (35)$$

Again, as in the three time level scheme, only η^* and U^* are solved in the x-sweep and only η^{n+1} and v^{n+1} are solved in the y-sweep.

5.3. Numerical Stability. Implicit methods are characterized by a property of unconditional stability in the linear sense. The scheme used in WIFM as well as the fully implicit scheme used in the subject three-dimensional is limited by a weak condition, namely,

$$\Delta t \leq \min_{x,y} \left[\frac{(\Delta x, \Delta y)}{(u^2 + v^2)^{1/2}} \right] \quad (36)$$

This same criterion can be expected to apply to the internal mode computations since it is based on the largest horizontal convection speed. In general, this limitation on Δt is at least two orders of magnitude larger than the limit imposed on an explicit scheme by the surface gravity wave, namely,

$$\min_{x,y} \left[\frac{(\Delta x, \Delta y)}{(u^2 + v^2)^{1/2}} \right] \gg \min_{x,y} \left[\frac{\Delta x, \Delta y}{\sqrt{gd}} \right] \quad (37)$$

6. APPLICATIONS. Program WIFM has been used successfully in many applications conducted at WES. These include tidal circulation studies for Masonboro, Inlet, North Carolina (Butler and Raney, 1976), Coos Bay Inlet-South Slough, Oregon (Butler, 1978b); storm surge applications for Hurricane Eloise, Panama City, Florida (Butler and Wanstrath, 1976), Hurricane Carla, Galveston, Texas (Butler, 1978c), and Hurricane Betsy and Camille, Lake Pontchartrain, Louisiana (WES Technical Report to be published); tsunami inundation simulations for Crescent City, California (Houston and Butler, 1979) and the Hawaiian Islands (Houston, et al., 1977). A recent paper (Butler, 1980) summarizes these applications.

To exemplify use of the model a brief description of computational sensitivity to modeling assumptions for a Louisiana coastal storm surge investigation is presented. As part of a study of a hurricane barrier protection plan for Lake Pontchartrain, a northern boundary for the city of New Orleans, Louisiana, an open-coast storm surge model of the pertinent coastal region was developed. Figure 3 displays the computational grid used in the investigation. Still water depths reach 3,000 m in the south-eastern corner of the grid.

To insure the efficacy of all model assumptions tests were made with varying grid limits, time steps, and still-water depth limitations (usually made when running explicit formulated models to relax stability criterion restrictions on the computational time step). Six grids were formed by considering two seaward boundaries and three eastern lateral boundaries noted in Figure 3. Five separate cutoff depths were selected: 90, 240, 400, 550, and 3,000 m. The full set of runs was thus thirty in number. Various time steps were selected for a limited set of runs and the only effect noted was the typical erosion of numerical accuracy with increasing timestep. Table 1 displays peak surge elevation results for eighteen runs

and two selected gages (locations shown on Figure 3). Hydrograph comparison (observed vs largest grid/actual topography and smallest grid/90 m cutoff depth) for a gage at Biloxi, MS, is shown in Figure 4. Coastline peak surge behavior for largest grid/actual topography, largest grid/90 m cutoff depth and smallest grid/90 m cutoff depth is compared in Figure 5.

These results are essentially self-explanatory. What is demonstrated is that model users must assure themselves that assumptions made in model formulation are not affecting the numerical results. The model region must be properly selected and the deep shelf southeast of the Mississippi River Delta properly simulated. Inclusion of deep water in the topography suggests the appropriateness of an implicit model, particularly if fine resolution is required.

For a second example attention is drawn to the investigation of the dynamic response of coastal waters in the vicinity of Mississippi Sound. A current WES work unit calls for the application of WIFM to study the hydrodynamics and horizontal salinity gradient in the Sound. The model area and grid are shown in Figure 6. Seaward boundary conditions are provided via a Gulf tide model (Reid and Whitaker, 1982). This same grid also will be used for a realistic test of the subject three-dimensional model. In addition, to investigate the three-dimensional hydrodynamics on the model grid, the currents will be used as input to a sediment transport model to study the transport of sediments in the vicinity of the Sound. Applications of the three-dimensional model are described in Sheng and Butler (1982).

7. CONCLUSIONS AND RECOMMENDATIONS. This paper presents details of the development of a two-dimensional finite difference hydrodynamic model. Implementation of a two time level ADI algorithm to treat the external mode of a three-dimensional model is also briefly discussed. Coordinate transformations are used to obtain finer resolution in important local areas without sacrificing economical application of the models. References are given for specific investigations of the two-dimensional model along with an example of model sensitivity to parameterization. An obvious extension of the models discussed herein (as demonstrated by the conference keynote speakers) would be the implementation of boundary fitted coordinates via the use of elliptic grid generation techniques.

8. ACKNOWLEDGEMENT. The research described and example applications presented herein were conducted under various programs of the United States Corps of Engineers by the Waterways Experiment Station. Permission was granted by the Chief of Engineers to publish this information.

REFERENCES

1. Butler, H. L., 1978a. "Numerical Simulation of Tidal Hydrodynamics: Great Egg Harbor and Corson Inlets, New Jersey," Technical Report H-78-11, U.S. Army Waterways Experiment Station, CE, Vicksburg, MS, June 1978.
2. Butler, H. Lee, 1978b. "Numerical Simulation of the Coos Bay-South Slough Complex," Technical Report H-78-22, U.S. Army Engineer Waterways Experiment Station, CE, Vicksburg, MS, December 1978.
3. Butler, H. Lee, 1978c. "Coastal Flood Simulation in Stretched Coordinates," 16th International Conference on Coastal Engineering, Proc. to be published, ASCE, Hamburg, Germany, 27 August-1 September 1978.
4. Butler, H. L. and Raney, D. C., 1976. "Finite Difference Schemes for Simulating Flow in an Inlet-Wetlands System," Proceedings of the Army Numerical Analysis and Computers Conference, The Army Mathematics Steering Committee, Durham, NC, March 1976.
5. Butler, H. Lee and Wanstrath, J. J., 1976. "Hurricane Surge and Tidal Dynamic Simulation of Ocean Estuarine Systems," ASCE 1976 Hydraulics Division Specialty Conference, Purdue University, Indiana, 4-6 August 1976.
6. Butler, H. Lee, "Evolution of a Numerical Model for Simulating Long-Period Wave Behavior in Ocean-Estuarine Systems," Estuarine and Wetland Processes with Emphasis on Modeling, Marine Science Series, Volume 11, Plenum Press, New York, 1980.
7. Houston, J. R., Carver, R. D., and Markle, D. G., 1977. "Tsunami-Wave Elevation Frequency of Occurrence for the Hawaiian Islands," Technical Report H-77-16, U.S. Army Engineer Waterways Experiment Station, CE, Vicksburg, MS, 1977.
8. Houston, James R. and Butler, H. Lee, 1979. "A Numerical Model for Tsunami Inundation," Technical Report HL-79-2, U.S. Army Engineer Waterways Experiment Station, CE, Vicksburg, MS, February 1979.
9. Leendertse, J. J., 1970. "A Water-Quality Simulation Model for Well-Mixed Estuaries and Coastal Seas, Vol. 1, Principals of Computation," RM-6230-rc, Rand Corp., Santa Monica, CA, February 1970.
10. Reid, R. O. and Whitaker, R. E., 1982. "Numerical Model for Astronomical Tides in the Gulf of Mexico, Vol. 1, Theory and Application," WES Contract Report in publication.
11. Sheng, Y. P., 1975. "Wind-Driven Currents and Dispersion of Contaminants in the Near-Shore Regions of Large Lakes," Report H-75-1, Waterways Experiment Station, CE, Vicksburg, MS.

12. Sheng, Y. P., W. Lick, R. Gedney, and F. Molls, 1978. "Numerical Computation of the Three-Dimensional Circulation in Lake Erie; A Comparison of a Free-Surface and A Rigid-Lid Model," J. Phys. Oceano., Volume 8, pp. 713-727.
13. Sheng, Y. P., 1980. "Modeling Sediment Transport in a Shallow Lake," Estuarine and Wetland Processes with Emphasis on Modeling, Marine Science Series, Volume 11, Plenum Press, New York.
14. Sheng, Y. P., 1981. "Modeling the Hydrodynamics and Dispersion of Sediments in the Mississippi Sound," A.R.A.P. Report No. 455, 107 pp.
15. Sheng, Y. P. and Butler, H. L., 1982. "A Three-Dimensional Hydrodynamic Model for Coastal, Estuarine, and Lake Currents," Proceedings of 1982 Army Numerical Analysis and Computers Conference, Vicksburg, Mississippi, 3 and 4 February 1982.

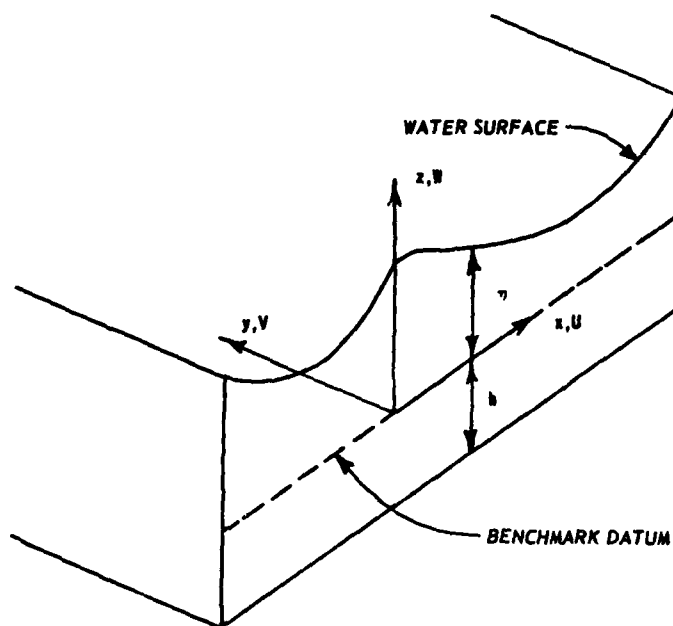


Figure 1. Cartesian coordinate system

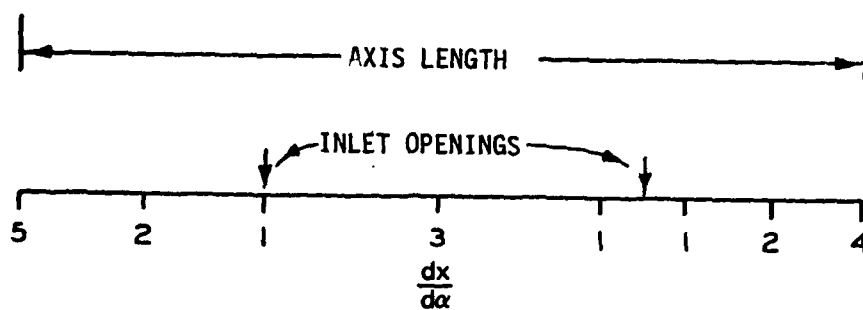


Figure 2. Illustration of coordinate stretching rates along a region axis

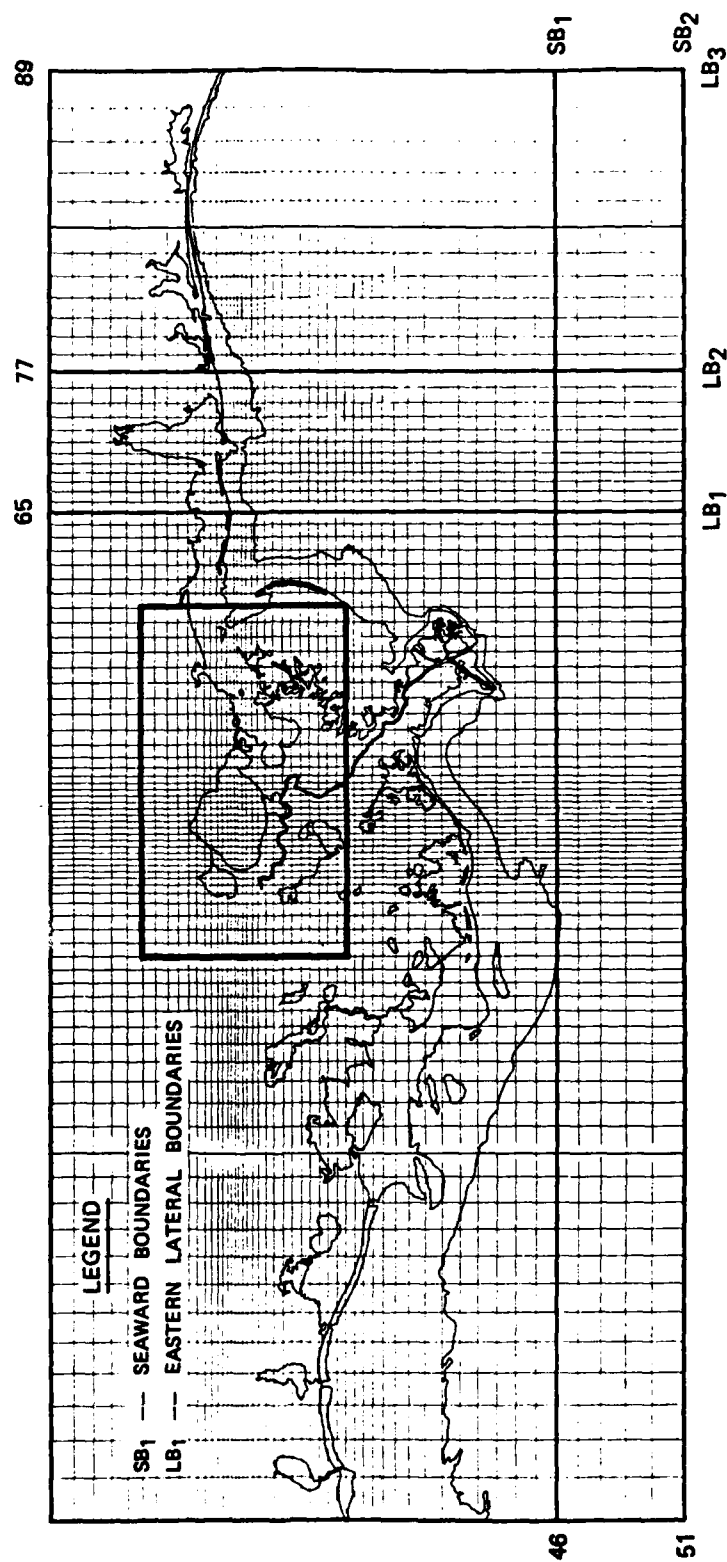


Figure 3. Computational grid for hurricane surge simulation in the vicinity of Lake Pontchartrain, LA

TABLE 1. PEAK SURGE IN METERS FOR MODEL GAGES AT
BILOXI, MS AND GRAND ISLE, LA

DEPTH LIMITATION (m) (1)	GRID DIMENSIONS					
	89,51 (2)	89,46 (3)	77,51 (4)	77,46 (5)	65,51 (6)	65,46 (7)
(a) BILOXI, MS						
90	2.7	2.7	2.3	2.3	1.8	1.8
240	2.5	2.5	2.3	2.3	1.8	1.8
3000	2.4	2.4	2.3	2.3	1.8	1.8
(b) GRAND ISLE, MS						
90	2.4	1.9	2.3	1.9	2.1	1.8
240	2.2	1.8	2.1	1.8	2.0	1.8
3000	1.9	1.8	1.9	2.0	1.9	1.8

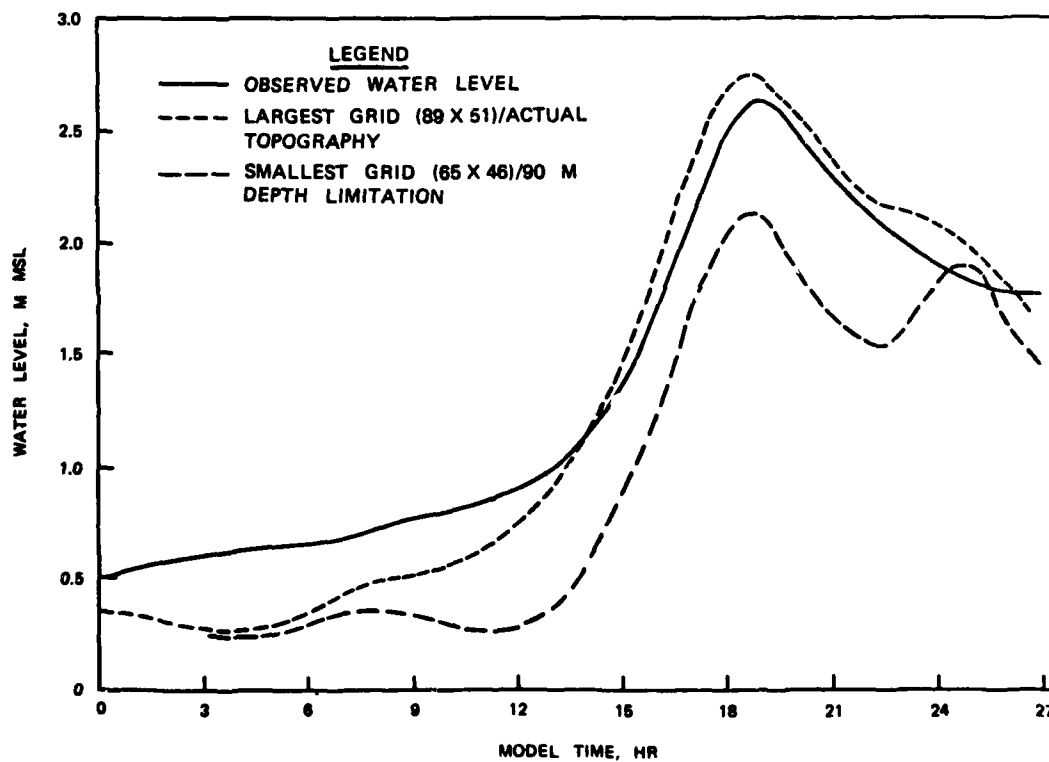


Figure 4. Comparison of computer water levels at
Biloxi, Mississippi vs observed levels

AD-A118 920 ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC F/8 12/1
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82
UNCLASSIFIED ARO-82-3 NI

ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC F/O 12/1
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82
ARO-82-3
NL

F/G 12/1

UNCLASSIFIED ARO-82-3

NL

50. 7

1

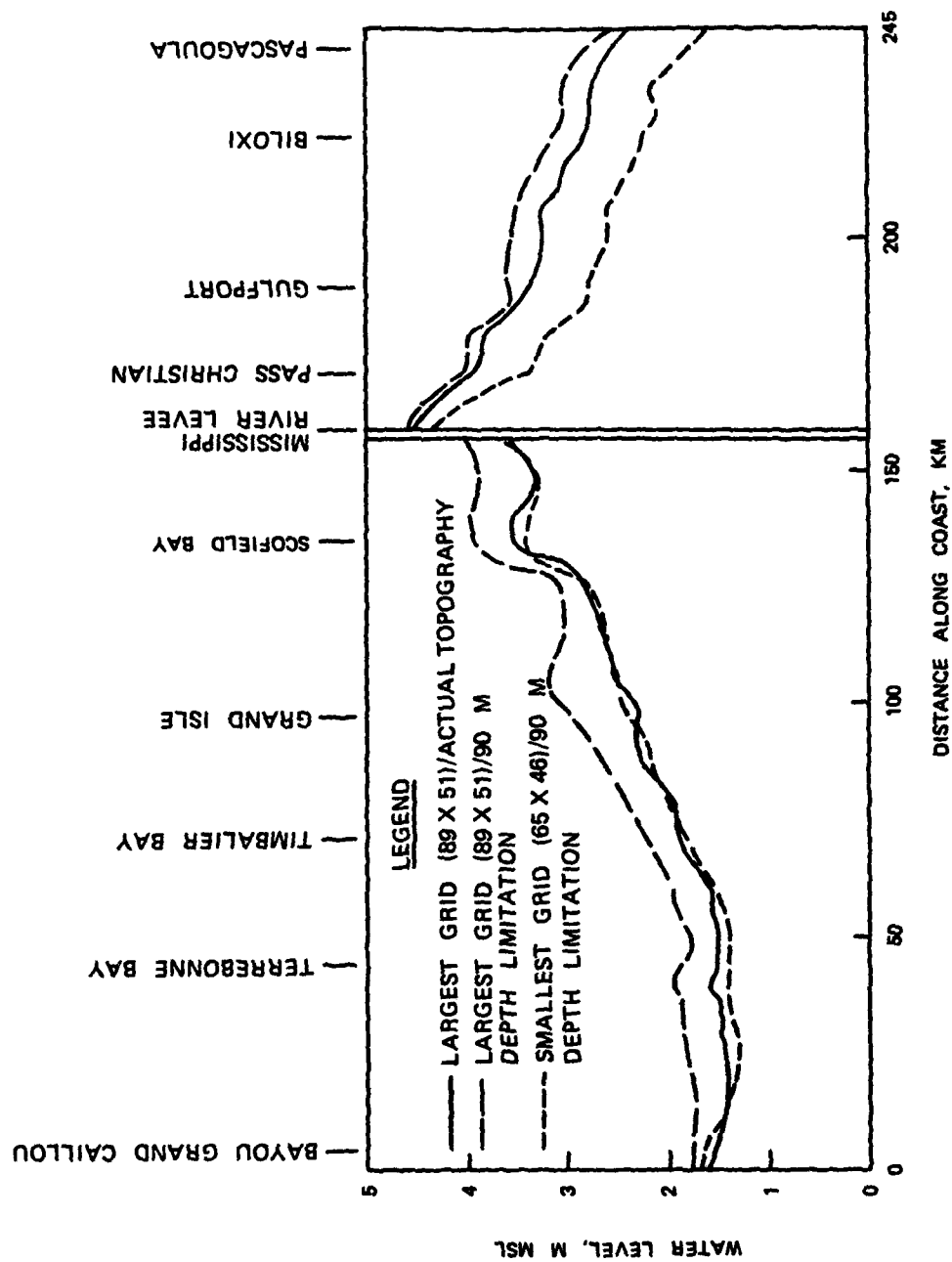


Figure 5. Comparison of computer peak water levels at the coast for various grid dimensions and depth limitations

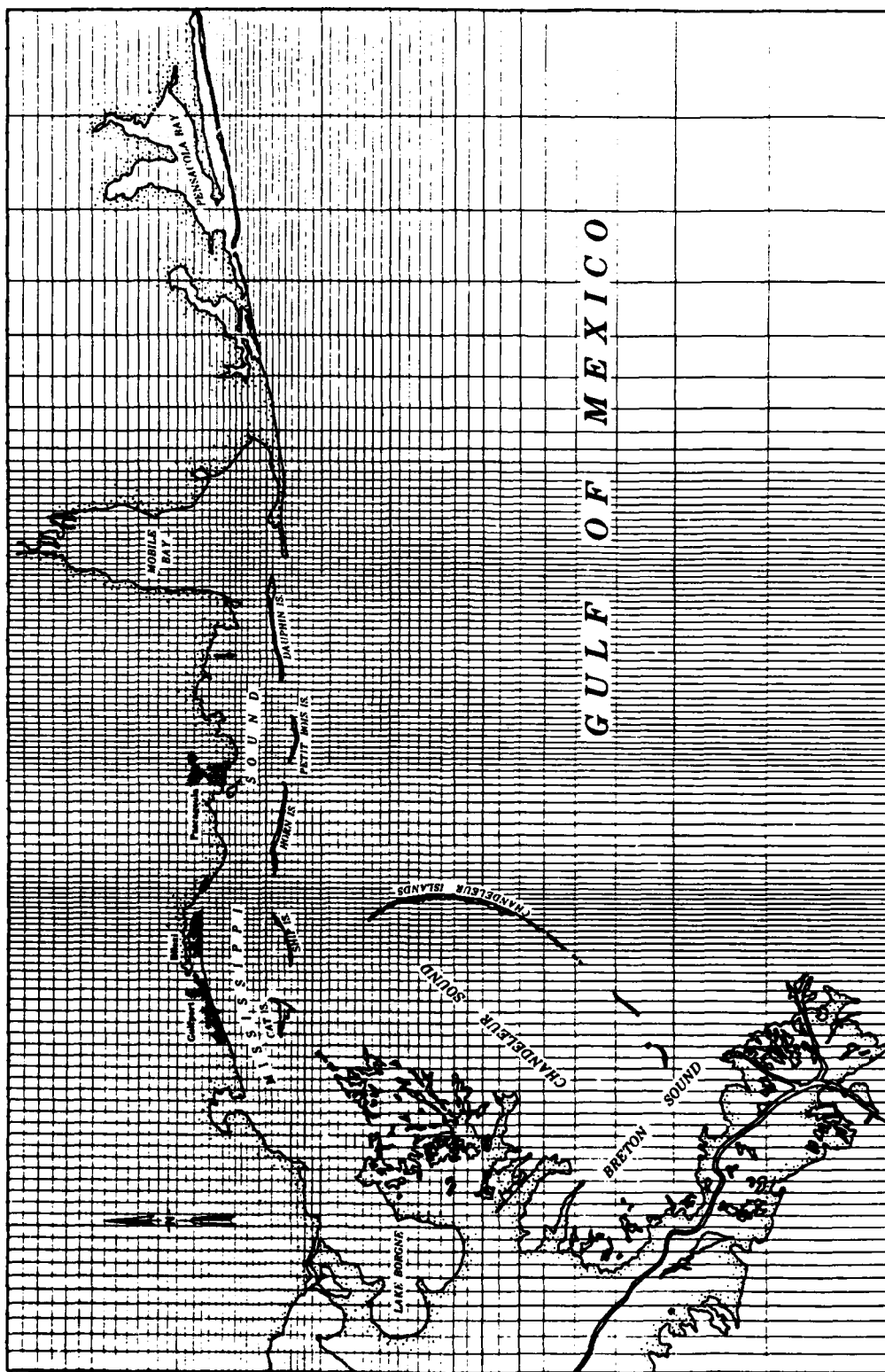


Figure 6. Computational grid for two- and three-dimensional hydrodynamic simulations in the vicinity of Mississippi Sound

ASYMPTOTIC AND NUMERICAL METHODS FOR VECTOR SYSTEMS
OF SINGULARLY-PERTURBED BOUNDARY VALUE PROBLEMS*

Joseph E. Flaherty
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12181

and

U.S. Army Armament Research and Development Command
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189

and

Robert E. O'Malley, Jr.
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12181

ABSTRACT. Procedures are developed for constructing asymptotic solutions for certain nonlinear singularly-perturbed vector two-point boundary value problems having boundary layers at one or both end points. The asymptotic approximations are generated numerically and can either be used as is or to furnish a two-point boundary value code (e.g. COLSYS) with an initial approximation and a nonuniform computational mesh. The procedures are applied to several examples involving the deformation of nonlinear elastic beams.

1. INTRODUCTION. We consider singularly-perturbed two-point boundary value problems for nonlinear vector systems of the form

$$\dot{x} = f(x, y, t, \epsilon) \quad , \quad \epsilon \dot{y} = g(x, y, t, \epsilon) \quad , \quad 0 < t < 1 \quad (1a, b)$$

$$a(x(0), y(0), \epsilon) = 0 \quad , \quad b(x(1), y(1), \epsilon) = 0 \quad (1c, d)$$

where x , y , a , and b are vectors of dimension m , n , q , and $r = m + n - q$, respectively. We seek to find limiting solutions of problem (1) as the small positive parameter ϵ tends to zero; however, to do this in complete generality is very difficult and beyond the grasp of our current understanding. Thus, we

*This research was partially sponsored by the U.S. Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant Number AFOSR 80-0192 and by the Office of Naval Research under Contract Number N00014-81K-056. The United States Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

simplify problem (1) considerably by assuming, in addition to natural smoothness hypothesis, that (i) g , a , and b are linear functions of the fast variable y , i.e.

$$g(x, y, t, \epsilon) = g_1(x, t, \epsilon) + G_2(x, t, \epsilon)y \quad (2a)$$

$$a(x(0), y(0), \epsilon) = a_1(x(0), \epsilon) + A_2(x(0), \epsilon)y(0) \quad (2b)$$

$$b(x(1), y(1), \epsilon) = b_1(x(1), \epsilon) + B_2(x(1), \epsilon)y(1) \quad (2c)$$

(ii) that $G_2(x, t, \epsilon)$ has a hyperbolic splitting with $k > 0$ stable eigenvalues and $n - k > 0$ unstable eigenvalues for all x and $0 < t < 1$, and (iii) that $q > k$ and $r > n - k$.

With the assumed hyperbolic splitting, we would expect y to vary rapidly relative to (the slow vector) x in narrow boundary layer regions near both $t = 0$ and 1 . We thus seek limiting solutions having the form

$$x(t, \epsilon) = X(t) + O(\epsilon) \quad , \quad y(t, \epsilon) = Y(t) + \mu(\tau) + v(\sigma) + O(\epsilon) \quad (3a, b)$$

where the initial layer correction $\mu(\tau)$ and the terminal layer correction $v(\sigma)$, respectively, decay to zero as the stretched variable

$$\tau = t/\epsilon \quad \text{or} \quad \sigma = (1-t)/\epsilon \quad (4a, b)$$

tend to infinity. The limiting solution $X(t)$, $Y(t)$ within $0 < t < 1$ must necessarily satisfy the reduced system

$$\dot{X} = f(X, Y, t, 0) \quad , \quad 0 = g(X, Y, t, 0) \quad (5a, b)$$

Because G_2 is everywhere nonsingular, we can use Eqs. (2a) and (5b) to determine

$$Y(t) = -G_2^{-1}(X, t, 0)g_1(X, t, 0) \quad (6)$$

in a locally unique way, and there remains the m th order nonlinear differential system (Eq. (5a)) for determining $X(t)$.

In order to completely specify the reduced solution we must prescribe m boundary conditions for equations (5a). We do this by providing a "cancellation law" which selects a combination of $q-k$ initial conditions (Eq. (2b)) and of $r - n + k$ terminal conditions (Eq. (2c)) to be satisfied by X and Y . In Section 2 we present a numerical procedure for determining the boundary conditions for the reduced system that uses an orthogonal matrix $E(x, t)$ to reduce the matrix $G_2(X(t), t, 0)$ to a block tridiagonal form so that the stable and unstable eigenspaces may be separated. The boundary layer corrections $\mu(\tau)$ and $v(\sigma)$ in Eqs. (3) compensate for the cancelled initial and terminal conditions, respectively, and they can be determined once $X(t)$ has been computed (cf. Section 2). This process avoids complicated matching procedures.

In Section 3 we discuss a numerical procedure for determining the asymptotic approximation (Eq. (3)) which uses the general purpose two-point boundary value code COLSYS to solve the reduced problem and then adds numerical approximations to the boundary layer corrections. This approximation is considerably less expensive to obtain than solving the full stiff problem numerically and it has the advantage of improving in accuracy, without any additional computational cost, as the small parameter ϵ tends to zero. However, when ϵ is only moderately small our asymptotic approximation may not be sufficiently accurate for some purposes, so we have developed a procedure (cf. Section 3) that generates an improved solution by using COLSYS to solve the complete problem (Eqs. (1) and (2)) with our asymptotic approximation as an initial guess. In order for this approach to succeed we must also provide COLSYS with an initial nonuniform mesh that is appropriately graded in the boundary layers (cf. Ascher and Weiss (Ref. 2)) and we give an algorithm for constructing such a mesh in Section 3. While our procedure does not appear to be optimal, we show by an example involving the deformation of a nonlinear elastic beam (cf. Section 4) that it does offer some advantage over the more standard approach of continuation in ϵ , where one starts with a large value of ϵ (e.g. $\epsilon = 1$) and a crude initial guess and reduces ϵ in steps so that the mesh is gradually concentrated into boundary layer regions.

We close Section 4 with a second nonlinear beam example that is beyond the capabilities of our present methods because the matrix G_2 is a function of y . Flaherty and O'Malley (Ref. 6) analyzed this problem and showed that its solution becomes unbounded as $\epsilon \rightarrow 0$. We include the numerical solution of this problem in this paper in order to show one of the many challenging effects that can occur with singularly-perturbed problems.

Finally, in Section 5 we discuss our results and present some suggestions for future investigations.

2. ASYMPTOTIC APPROXIMATION. In order to calculate the boundary conditions for the reduced problem (Eqs. (5a) and (6)) and the boundary layer corrections $\mu(\tau)$ and $v(\sigma)$ we calculate the Schur decomposition of the matrix G_2 at $t = 0$ and $t = 1$. In particular, at $t = 0$ we find an orthogonal matrix $E(x(0))$ such that

$$G_2(x(0), 0, 0)E(x(0)) = E(x(0)) \begin{bmatrix} T_-(x(0)) & U(x(0)) \\ 0 & T_+(x(0)) \end{bmatrix} \quad (7)$$

where T_- is $k \times k$ and upper triangular with the stable eigenvalues of G_2 , and T_+ is upper triangular with the $n-k$ unstable eigenvalues of G_2 . The decomposition (Eq. (7)) can often be obtained analytically; however, when this is not possible or practical it can be determined numerically by using the QR algorithm (cf. Golub and Wilkinson (Ref. 7) and Ruhe (Ref. 9) for specific procedures).

We partition E after its k th column as

$$E = [E_- \bar{E}_-] \quad (8)$$

and note that E_- spans the stable eigenspace of G_2 at $t = 0$ and

$$P = E_- E_-^T \quad (9)$$

is a projection onto this eigenspace.

Near $t = 0$, we assume that the terminal layer correction v is negligible, substitute the asymptotic approximation (Eq. (3)) into the differential equations (Eqs. (1a,b)), use the reduced system (Eq. (5)), and retain only the leading order terms to find that $\mu(\tau)$ satisfies the conditionally stable system

$$d\mu/d\tau = G_2(0)\mu \quad (10)$$

where (here and below) we use the argument t to denote conditions evaluated at $x(t) = X(t)$, t , and $\epsilon = 0$, e.g.,

$$G_2(0) := G_2(X(0), 0, 0) \quad (11)$$

Integrating Eq. (10)

$$\mu(\tau) = e^{G_2(0)\tau} \mu(0) \quad (12)$$

We require that $\mu(\cdot)$ decays as τ increases and this will be the case provided that $\mu(0)$ is in the stable eigenspace of $G_2(0)$; thus, using Eq. (9) we require

$$\mu(0) = P(0)\mu(0) = E_-(0)E_-^T(0)\mu(0) \quad (13)$$

Using Eqs. (3), (13), and (2b) in Eq. (1b) we find that the limiting initial conditions have the form

$$a_1(0) + A_2(0) [Y(0) + E_-(0)E_-^T(0)\mu(0)] = 0 \quad (14)$$

We assume that $A_2(0)E_-(0)$ has its maximal rank k and construct a $q \times q$ matrix

$$L^T = [L_-^T \bar{L}_-^T] \quad (15a)$$

that reduces it to row echelon form, i.e.,

$$\begin{bmatrix} L_- \\ - \\ L_- \end{bmatrix} A_2(0)E_-(0) = \begin{bmatrix} V_- \\ 0 \end{bmatrix} \quad (15b)$$

where V_- is $k \times k$ and nonsingular. Multiplying Eq. (14) by L and using Eqs. (13) and (15) gives the initial layer jump and the $q-k$ initial conditions for the reduced problem, respectively, as

$$u(0) = -E_-(0)V_-^{-1}L_-[a_1(X(0),0) + A_2(X(0),0)Y(0)] \quad (16a)$$

and

$$\phi(X(0)) := \bar{L}_-[a_1(X(0),0) + A_2(X(0),0)Y(0)] = 0. \quad (16b)$$

We find the terminal layer jump and the $r - (n-k)$ terminal conditions for the reduced problem in an analogous fashion with the exception that we define $E(x(1))$ such that

$$G_2(x(1),1,0)E(x(1)) = E(x(1)) \begin{bmatrix} \hat{T}_+(x(1)) & \hat{U}(x(1)) \\ 0 & \hat{T}_-(x(1)) \end{bmatrix} \quad (17)$$

which we partition after its $(n-k)$ th column as

$$E = [E_+ \quad \bar{E}_+] \quad (18)$$

In parallel with Eqs. (7) and (8), the matrices \hat{T}_- , \hat{T}_+ , and E_+ contain the k stable eigenvalues, the $n-k$ unstable eigenvalues, and span the unstable eigenspace, respectively, of G_2 at $t = 1$. Our reasons for switching the positions of the matrices containing the stable and unstable eigenvalues of G_2 is that there is no simple and stable computational procedure for finding a set of vectors that span a given subspace and are not in the leading columns of an orthogonal matrix like E (cf. Golub and Wilkinson (Ref. 7)).

Now, following the procedure that we used for the initial layer, we find that the terminal layer correction satisfies

$$v(\sigma) = e^{G_2(1)\sigma} v(0) \quad (19)$$

In order for $v(\sigma)$ to decay as σ increases we require $v(0)$ to be in the unstable eigenspace of $G_2(1)$; thus, we take

$$v(0) = Q(1)v(0) = E_+(1)E_+^T(1)v(0) \quad (20)$$

where Q is a projection onto the $(n-k)$ dimensional unstable eigenspace of $G_2(1)$.

We assume that $B_2(1)E_+(1)$ has its maximal rank $n-k$ and find a $r \times r$ matrix

$$R^T = [R_+^T \quad \bar{R}_+^T] \quad (21a)$$

that reduces it to the row echelon form

$$\begin{bmatrix} R_+ \\ \bar{R}_+ \end{bmatrix} B_2(1)E_+(1) = \begin{bmatrix} V_+ \\ 0 \end{bmatrix} \quad (21b)$$

where V_+ is $(n-k) \times (n-k)$ and nonsingular. Multiplying Eq. (1d) by R , using Eqs. (2c), (3), (20), and (21), and retaining only the leading order terms we find the terminal layer jump and the $r \rightarrow (n-k)$ terminal conditions for the reduced problem, respectively, as

$$v(0) = -E_+(1)V_+^{-1}R_+[b_1(X(1),0) + B_2(X(1),0)Y(1)] \quad (22a)$$

and

$$\Psi(X(1)) := R_+[b_1(X(1),0) + B_2(X(1),0)Y(1)] = 0 \quad (22b)$$

In the interest of brevity, we have omitted several details of our construction and have not attempted to justify the asymptotic validity of our procedure. These topics will be the subject of a forthcoming paper by O'Malley and Flaherty (Ref. 8).

3. NUMERICAL PROCEDURE. Our computational procedure consists of first solving the reduced problem (cf. Eqs. (5a), (6), (16b), and (22b)) numerically and then adding any boundary layer corrections. Since the reduced problem is not stiff we can use any good code for two-point boundary value problems (cf. Childs et al. (Ref. 3)) and we have chosen to use the collocation code COLSYS of Ascher, Christiansen, and Russell (Ref. 1).

Since the reduced problem is generally nonlinear and since COLSYS solves nonlinear problems using a damped Newton method we have to supply formulas for evaluating the Jacobians of f , Y , Φ , and Ψ with respect to X . We do this by providing analytical formulas for these Jacobians that neglect the influence of the derivatives of E , L , R , and G_2 . This procedure has not failed on any of our examples; however, an alternate possibility would be to approximate the Jacobians by finite differences.

We start the Newton iteration with a uniform mesh and the default initial guess $X^{(0)}(t)$ for $X(t)$ that is provided by COLSYS and calculate successive approximations $X^{(p)}(t)$ until convergence is attained. At each iteration step we calculate an approximation $E^{(p)}(t)$ to $E(t)$ for $t = 0$ and 1 as the Schur decomposition of $G_2(X^{(p)}(t), t, 0)$. In the examples of Section 4 we used analytical formulas for E rather than the numerical procedures of Golub and Wilkinson (Ref. 7) or Ruhe (Ref. 9). Finally, $L^{(p)}$ and $R^{(p)}$ are obtained using Gaussian elimination to row reduce $A_2(X^{(p)}(0), 0)E_-^{(p)}(0)$ and $B_2(X^{(p)}(1), 0)E_+^{(p)}(1)$, respectively.

When the above procedure converges we calculate boundary layer corrections $u(\tau)$ and $v(\sigma)$, for a given value of ϵ , using Eqs. (12), (16a), (19), and (22a), and add these to the reduced solution in order to get the $O(\epsilon)$ asymptotic approximation (Eq. (3)). For moderately small values of ϵ this approximation may not provide a sufficiently accurate representation of the solution and, in this case, we use it as an initial guess to COLSYS and solve the complete problem (Eq. (1)). Unfortunately, this procedure will fail unless we also provide COLSYS with an initial nonuniform partition

$$\pi := \{0 = t_0 < t_1 < \dots < t_N = 1\} \quad (23)$$

that is appropriately graded within the boundary layers. We seek to find π so that the pointwise error satisfies

$$||e(t_i)|| < \delta(1 + ||u(t_i)||) \quad , \quad i = 1, 2, \dots, N-1 \quad (24)$$

where δ is a prescribed tolerance, $u^T := [x^T, y^T]$, e is the difference between u and its collocation approximation, and

$$||u(t_i)|| := \max_{1 \leq j \leq m+n} |u_j(t_i)| \quad (25)$$

We have based our condition for determining π on a pointwise error criteria since this seemed to work better in practice than a global criteria. This is somewhat surprising since COLSYS uses a global error criteria to select a mesh.

We assume that the final partition selected by COLSYS to solve the reduced problem satisfies equation (24) outside of boundary layer regions and we seek to refine it within the boundary layers. We further assume that derivatives of u can adequately be replaced by either $u(\tau)$ or $v(\sigma)$ in the left or right boundary layer, respectively.

This problem was studied by Ascher and Weiss (Ref. 2) who showed that Eq. (24) could be approximately satisfied in the left boundary layer by choosing subinterval lengths as

$$t_i - t_{i-1} = \left(\frac{\epsilon}{\alpha} \right) \left[\frac{\delta(1 + ||u(t_{i-1})||)^{1/2k}}{c ||u(t_{i-1})||} \right] \quad (26)$$

for collocation at the image of k Gauss-Legendre points per subinterval. Here c is a numerical constant and α is the magnitude of the largest diagonal element of $T_-(X(0))$. A similar formula can be obtained for selecting subinterval lengths in the right boundary layer.

Starting with $i = 1$ we use Eq. (26) to generate a partition until we either reach $t = 1/2$ or a point where a subinterval length selected by Eq. (26) is larger than that used by COLSYS to solve the reduced problem. We then repeat the procedure in the right boundary layer.

We have written a computer code called SPCOL that implements the algorithms that are described in this section; thus, it (i) uses COLSYS to solve the reduced problem, (ii) calculates and adds appropriate boundary layer corrections to the reduced problem, and (iii) (optionally) suggests a mesh that can be used by COLSYS to solve the complete problem.

4. EXAMPLES. In order to appraise the performance of SPCOL we have applied it to several examples involving the deformation of a nonlinear elastic beam which is resting on a nonlinear elastic foundation and is subjected to the combined action of a horizontal end thrust P and a lateral load $p(x,t)$ per unit length (cf. Figure 1). This problem is discussed and analyzed in detail in Flaherty and O'Malley (Ref. 6) and herein we only present the governing equations, which in dimensionless form are

$$\dot{x}_1 = \cos x_3, \quad \dot{x}_2 = \sin x_3, \quad \dot{x}_3 = y_1 \quad (27a,b,c)$$

$$\epsilon \dot{y}_1 = -y_2, \quad \epsilon \dot{y}_2 = (\lambda^2 x_2 - p) \cos x_3 - T y_1, \quad (27d,e)$$

where

$$T = \sec x_3 + \epsilon y_2 \tan x_3 \quad (27f)$$

The slow variables (x_1, x_2) and x_3 represent the Cartesian coordinates and the tangent angle of a material particle on the centerline of the beam that was at the Cartesian location $(t, 0)$ in the undeformed state. The fast variables y_1 and y_2 are the internal bending moment and transverse shear force, respectively (cf. Figure 1). Finally, the small parameter is

$$\epsilon^2 = EI/PL^2, \quad (28)$$

where EI is the flexural rigidity and L is the length of the beam; thus, our beam is much stronger in extension than it is in bending.

This example does not precisely fit out hypotheses since the axial force T is a function of the fast variable y_2 and, thus, G_2 also depends on y . However, our theory and methods will still apply as long as y remains bounded and $|x_3| < \pi/2$ as $\epsilon \rightarrow 0$. In order to illustrate the diverse behaviors that can occur when y either does or does not remain bounded as $\epsilon \rightarrow 0$ we present solutions for two problems both having $\lambda = p = 1$ and which differ only in their boundary conditions. Some additional examples are presented in Flaherty and O'Malley (Refs. 6 and 8).

In our first example we take the boundary conditions as

$$\begin{aligned} x_1(0) = 0, \quad -10x_2(0) + y_2(0) = 0, \quad -x_3(0) + 10y_1(0) = 0 \\ 10x_2(1) + y_2(1) = 0, \quad 10x_3(1) + y_1(1) = 0 \end{aligned} \quad (29)$$

These supports correspond to a beam that is almost simply supported at $t = 0$ and almost clamped at $t = 1$. However, perhaps due to friction, there is some coupling between lateral and rotational effects at the supports.

As we shall see, y remains bounded in this example so our methods are applicable. The orthogonal matrix

$$E(x(0)) = (1 + \alpha^2)^{-1} \cdot \begin{bmatrix} 1 & -\alpha \\ \alpha & 1 \end{bmatrix} \quad (30a)$$

where

$$\alpha^2 = \sec x_3(0) \quad (30b)$$

reduces

$$G_2(x(0), 0, 0) = \begin{bmatrix} 0 & -1 \\ -\alpha^2 & 0 \end{bmatrix} \quad (31)$$

to the Schur form given by equation (7) at $t = 0$ and E^T will reduce $G_2(x(1), 1, 0)$ to the form given by Eq. (17) at $t = 1$.

We solved this problem in two ways: (i) using COLSYS to solve the complete problem (Eqs. (27) and (29)) with continuation from a large to a small value of ϵ and (ii) using our code SPCOL to compute an initial asymptotic approximation and to recommend a nonuniform mesh and using this with COLSYS to calculate an improved solution. All calculations were performed in double precision on an IBM 3033 computer, used two collocation points per subinterval, and set the error tolerance δ (cf. Eq. (24)) at 10^{-3} for slow variables and 10^{-3} for fast variables.

Our results for the normalized CP times and the number of subintervals (NSUB) that are either used by COLSYS or recommended by SPCOL are shown in Tables 1 and 2 for continuation in ϵ and our methods, respectively. Differences between our initial asymptotic approximation and the final solution obtained by COLSYS are shown for x_3 and y_2 at $t = 0$ and 1 in Table 3. We see that the differences decrease like $O(\epsilon)$ as expected. Differences that are recorded as zero are less than 10^{-8} . Finally, we exhibit solutions for x_2 , x_3 , y_1 , and y_2 in Figure 2.

The results reported in Tables 1 and 2 need some additional explanation. The number of subintervals and CP times used with continuation depended heavily on the ϵ sequence that was used. The results in Table 1 are about the best insofar as they gave the smallest total CP time for the sequence. In addition, COLSYS relies on the difference between solutions that are computed on two different partitions in order to estimate local errors. Thus, at a minimum, COLSYS would always double our suggested mesh. This is apparent in the results listed under the heading of "COLSYS Correction No. 1" in Table 2. In some sense these results are encouraging insofar as they indicate that our mesh selection strategy is doing about as well as it can, at least for $\epsilon < 10^{-2}$. However, it seems that fewer points should be necessary, so we tried giving COLSYS an initial mesh that consisted of every other point of our recommended mesh. This is clearly a risky strategy since collocation at the Gauss-Legendre points is known to be unstable unless the mesh is sufficiently fine in the boundary layers (cf. Ascher and Weiss (Ref. 2)). Our results using this are reported under the heading of "COLSYS Correction No. 2" in Table 2. Some improvement is noted for $\epsilon > 10^{-4}$; however, COLSYS failed to find a solution (within our prescribed limitations) when $\epsilon = 10^{-8}$.

In our second example we use the boundary conditions

$$\begin{aligned} x_1(0) = 0, \quad -x_2(0) + \epsilon y_2(0) = 0, \quad -x_3(0) + \epsilon^2 y_1(0) = 0 \\ x_2(1) + \epsilon y_2(1) = 0, \quad x_3(1) + \epsilon^2 y_1(1) = 0 \end{aligned} \quad (32)$$

If ϵ were set to zero then these boundary conditions would correspond to clamped supports at $t = 0$ and 1 . Since the limiting boundary conditions only involve the slow variables and since the slow vector x cannot generally satisfy all of them as $\epsilon \rightarrow 0$ we would expect the solution to have boundary layers in these components. This in turn will force the fast vector y to become unbounded like $O(1/\epsilon)$ at the endpoints! Thus, this problem does not have an asymptotic expansion having the form of Eq. (3); however, an appropriate asymptotic representation of a solution has been obtained by Flaherty and O'Malley (Ref. 6). We shall not repeat those results here, but in order to emphasize the diverse behavior that can occur in nonlinear singularly-perturbed problems, we present solutions for x_2 , x_3 , ey_1 , and ey_2 in Figure 3. These solutions were computed using COLSYS with continuation in ϵ .

5. DISCUSSION. We have obtained asymptotic approximations for a restricted class of nonlinear singularly-perturbed boundary value problems and have shown how to construct them numerically and use them to suggest a nonuniform mesh that may be used as input to a two-point boundary value code in order to calculate improved solutions. Clearly this approach offers some advantages over the more standard technique of continuation in ϵ steps; however, the picture is far from clear and several questions still remain as to how best to use asymptotic analysis in conjunction with numerical analysis.

As we have shown in our second example of Section 4, very diverse behavior in the solution of singularly-perturbed problems can result from seemingly minor changes in boundary conditions. Some phenomena cannot easily be predicted, so perhaps a sensible course to follow is to use asymptotic and numerical methods in tandem. For example, a rough numerical solution could be obtained for several values of ϵ which could then be used to suggest the form of an asymptotic solution. The asymptotic approximation could then be used to refine the numerical solution, and so on. It is also possible that singular perturbation theory could be used to construct special methods that are appropriate for specific problems as e.g., in Flaherty and Mathon (Ref. 4) and Ascher and Weiss (Ref. 2).

Throughout our discussion we have ignored the question of uniqueness. In general, multiple solutions can be expected and they must be coped with numerically. In Reference (5) we showed how asymptotic methods may be used to distinguish the different solutions and to provide initial guesses for a two-point boundary value code.

REFERENCES

1. U. Ascher, I. Christiansen, and R. D. Russell, "Collocation Software For Boundary Value ODE's," ACM Trans. Math. Software, **7** (1981), pp. 209-222.
2. U. Ascher and R. Weiss, "Collocation For Singular Perturbation Problems I: First Order Systems With Constant Coefficients," Tec. Rep. 81-2, Dept. Comp. Sci., University of British Columbia, 1981.

3. B. Childs, M. Scott, J. W. Daniel, E. Denman, and P. Nelson (Eds.), Codes for Boundary-Value Problems in Ordinary Differential Equations, Proceedings of a Working Conference, May 14-17, 1978, Lecture Notes in Computer Science, No. 76, Springer-Verlag, Berlin, 1979.
4. J. E. Flaherty and W. Mathon, "Collocation with Polynomial and Tension Splines for Singularly-Perturbed Boundary Value Problems," SIAM J. Sci. Stat. Comput., 1 (1980), pp. 260-289.
5. J. E. Flaherty and R. E. O'Malley, Jr., "On the Numerical Integration of Two-Point Value Problems For Stiff Systems of Ordinary Differential Equations," Boundary and Interior Layers - Computational and Asymptotic Methods, J. J. H. Miller, Editor, Boole Press, Dublin, 1980, pp. 93-102.
6. J. E. Flaherty and R. E. O'Malley, "Singularly-Perturbed Boundary Value Problems For Nonlinear Systems, Including a Challenging Problem For a Non-linear Beam," Proceedings, Conference on Singulare Störungstheorie mit Anwendungen, Oberwolfach, 1981.
7. G. H. Golub and J. H. Wilkinson, "Ill-Conditioned Eigensystems and The Computation of the Jordan Canonical Form," SIAM Review 18 (1976), pp. 578-619.
8. R. E. O'Malley and J. E. Flaherty, "Numerical Methods For Stiff Systems of Two-Point Boundary Value Problems," to appear.
9. A. Ruhe, "An Algorithm for Numerical Determination of the Structure of a General Matrix," BIT, 10 (1970), PP. 196-216.

TABLE 1. NONLINEAR ELASTICALLY SUPPORTED BEAM. NUMBER OF SUBINTERVALS (NSUB) AND CP TIMES USED TO SOLVE THE PROBLEM BY COLSYS WITH CONTINUATION IN ϵ . THE TOTAL CP IS THE ACCUMULATED TIME FOR THE ϵ SEQUENCE.

ϵ	NSUB	CP	Total CP
10^{-1}	80	8.0	8.0
10^{-2}	78	9.0	17.0
10^{-4}	78	19.5	36.5
10^{-6}	156	44.5	81.0
10^{-8}	100	19.0	100.0

TABLE 2. NONLINEAR ELASTICALLY SUPPORTED BEAM. NUMBER OF SUBINTERVALS (NSUB) AND CP TIMES TO SOLVE THE PROBLEM BY SPCOL AND OBTAIN AN IMPROVEMENT BY COLSYS. THE CP TIMES FOR SPCOL INCLUDE THE TIME TO CALCULATE THE REDUCED SOLUTION WHICH WAS 4.8 TIME UNITS. CORRECTION NO. 1 USES THE MESH THAT WAS RECOMMENDED BY SPCOL. CORRECTION NO. 2 USES A MESH THAT IS TWICE AS COARSE. THE TOTAL CP IS THE SUM OF THE TIMES FOR THE SPCOL AND COLSYS SOLUTIONS.

ϵ	SPCOL		COLSYS Correction No. 1			COLSYS Correction No. 2		
	Rec. No. of NSUB	CP	NSUB	CP	Total CP	NSUB	CP	Total CP
10^{-1}	40	4.9	100	12.0	16.9	80	12.1	16.9
10^{-2}	45	4.9	90	12.0	16.9	78	8.1	12.9
10^{-4}	54	4.9	108	16.9	21.8	66	9.2	14.1
10^{-8}	55	4.9	110	17.5	22.3			Failed

TABLE 3. NONLINEAR ELASTICALLY SUPPORTED BEAM. DIFFERENCES BETWEEN SPCOL AND COLSYS SOLUTIONS, WHERE $\Delta(\) := |(\)_{\text{SPCOL}} - (\)_{\text{COLSYS}}|$

ϵ	$\Delta x_3(0)$	$\Delta y_2(0)$	$\Delta x_3(1)$	$\Delta y_2(1)$
10^{-1}	3.3×10^{-1}	5.1×10^{-2}	6.8×10^{-1}	3.6×10^{-1}
10^{-2}	2.8×10^{-2}	6.6×10^{-3}	6.1×10^{-2}	3.9×10^{-2}
10^{-4}	2.7×10^{-4}	6.8×10^{-5}	6.1×10^{-4}	3.9×10^{-4}
10^{-8}	0	1.3×10^{-7}	0	0

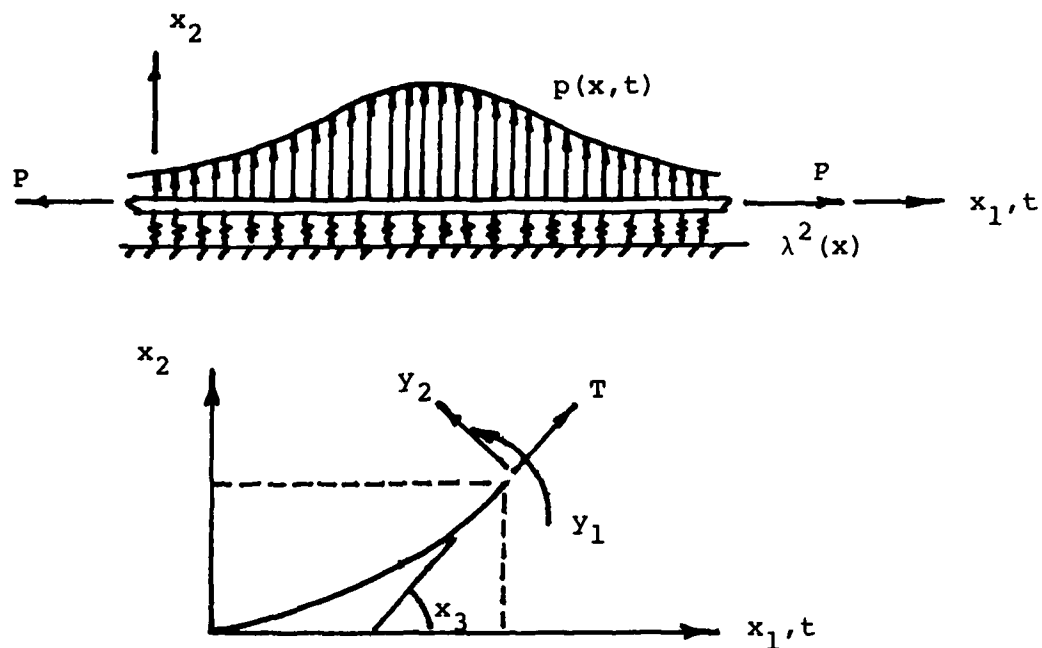


Figure 1. Geometry, loading, force, and moment conventions for nonlinear beam.

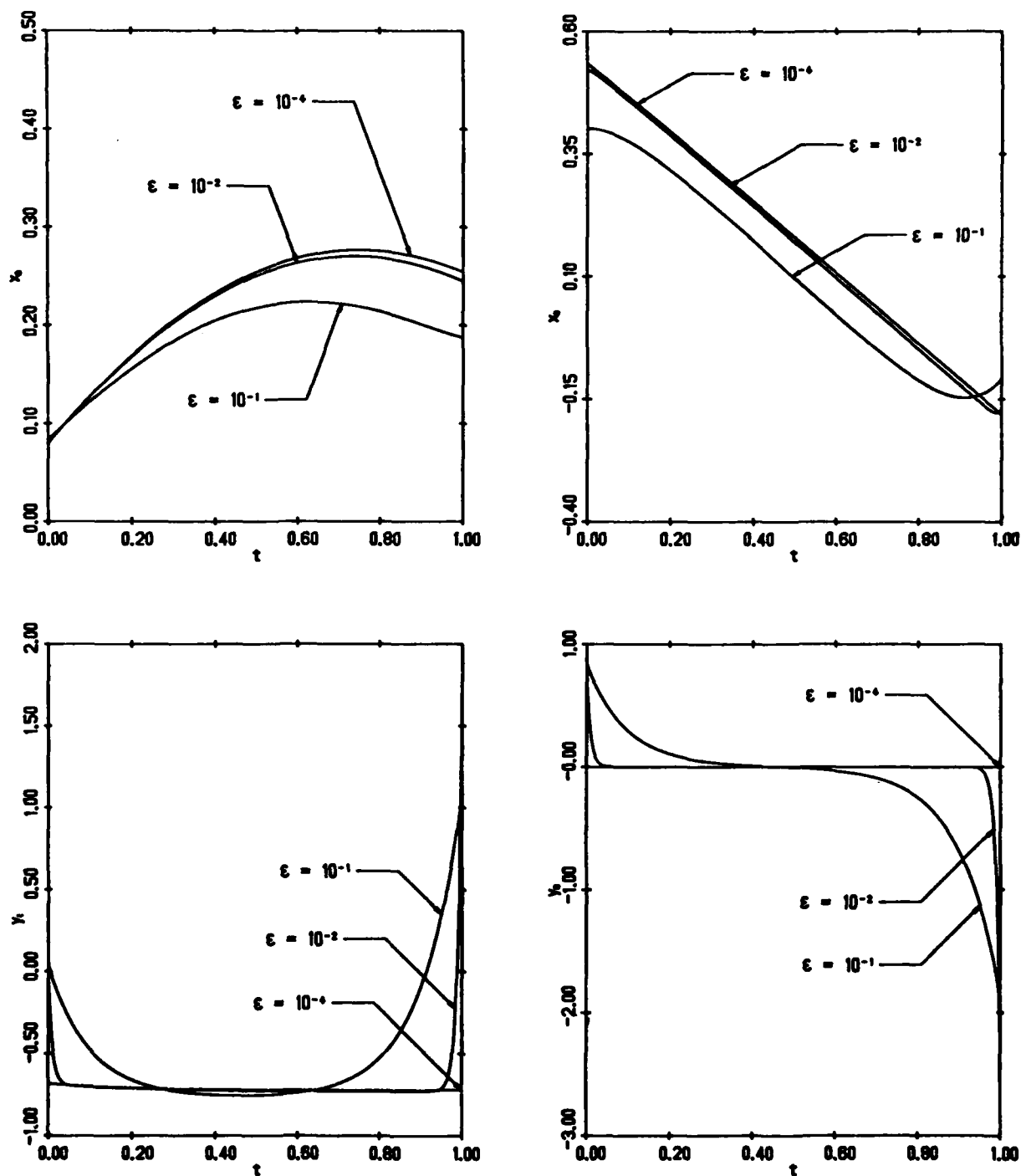


Figure 2. Numerical solution of elastically supported beam with boundary conditions given by Equations (29).

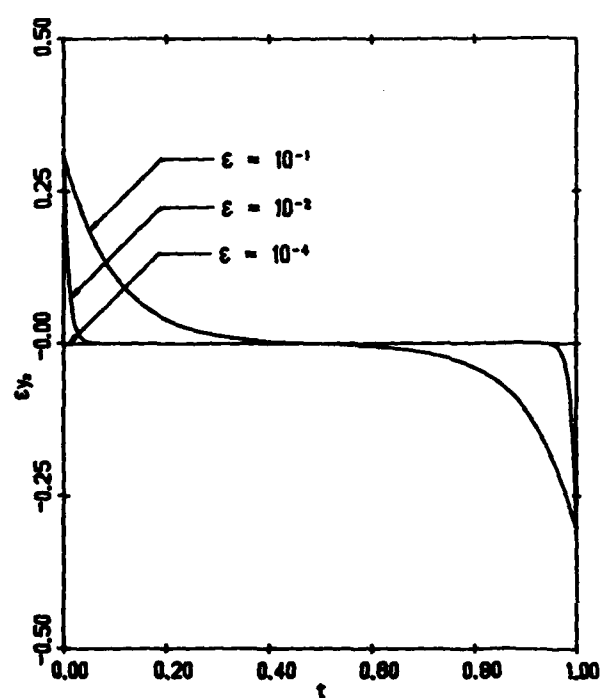
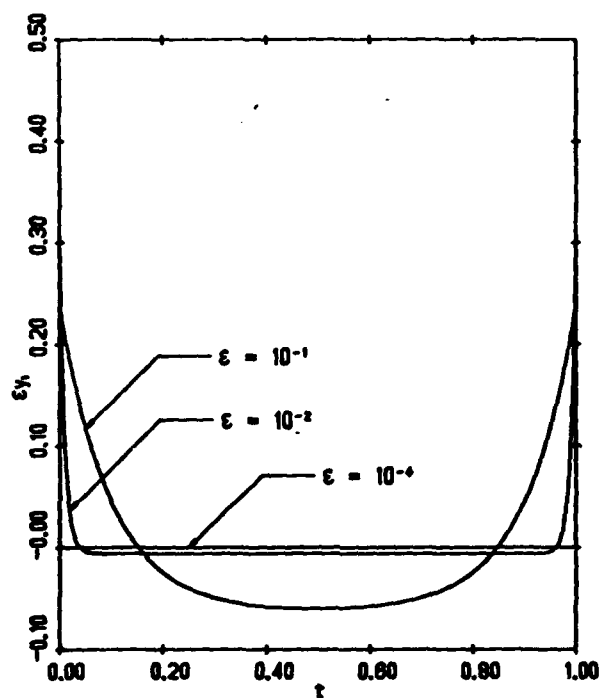
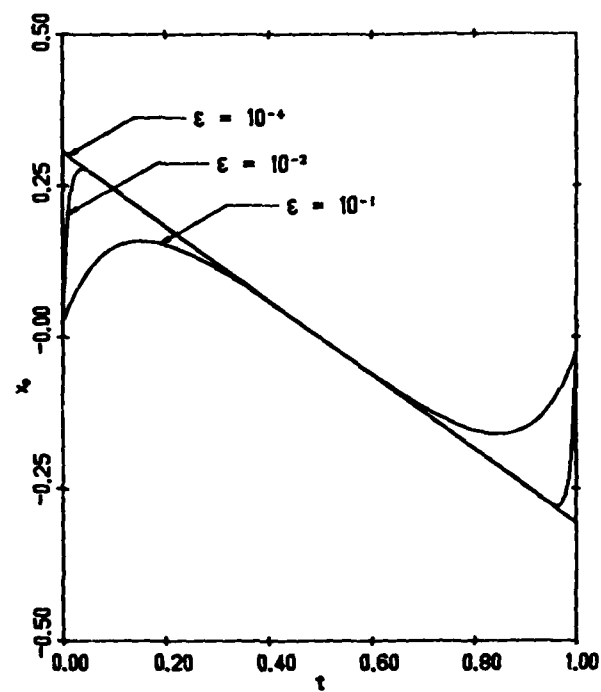
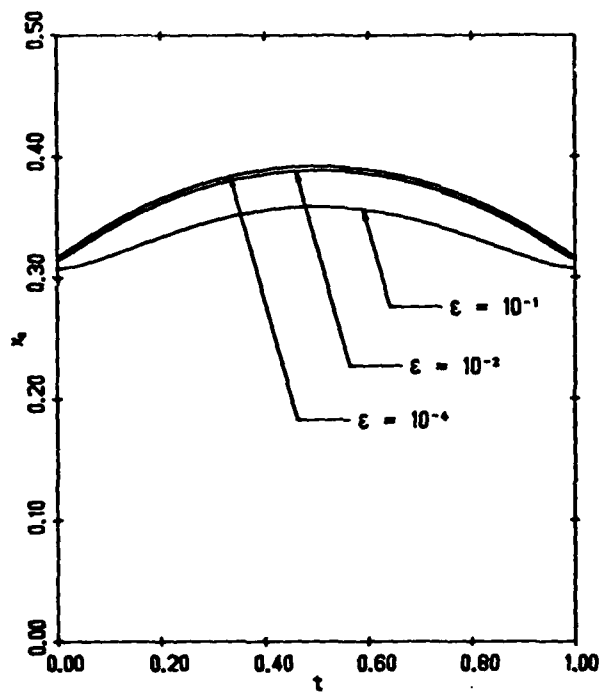


Figure 3. Numerical solution of elastically supported beam with boundary conditions given by Equations (32). Note that y_1 and y_2 are multiplied by ϵ .

AN INTEGRAL EQUATION FOR THE DESIGN OF MAGNETIC FIELD COILS

J.F. Schenck, M.A. Hussain,* W.A. Edelstein, B. Noble†
Electronics Systems Programs Operation
Corporate Research and Development
General Electric Company

INTRODUCTION

We have investigated the use of the calculus of variations to design magnet coils that have a minimum stored energy. In particular, we have considered the problem of computing the winding density along a cylindrical surface that will minimize the stored energy, while at the same time produce a magnetic field with certain desired characteristics. These desired characteristics are prescribed as constraints on various coefficients in the spherical harmonic expansion of the magnetic field. They are introduced into the minimization problem by the use of Lagrange multipliers.

This process leads to a linear integral equation of the form:

$$\int_{-Z_m}^{Z_m} Q(Z_o - Z'_o) \sigma_\phi(Z'_o) dZ'_o = f(Z_o), \quad -Z_m < Z_o < Z_m$$

with

$$f(Z_o) = \sum_n \lambda_n f_n(Z_o)$$

where a is the radius of the cylinder, $Z_o = z_o/a$ is the normalized position of the winding element along the cylindrical axis, $\sigma(Z_o)$ is the unknown winding density, Z_m is the half-length of the coil, the λ_n are Lagrange multipliers, and the $f_n(Z_o)$ are prescribed functions.

The kernel $Q(Z_o - Z'_o)$ is symmetric and has a logarithmic singularity at $Z_o = Z'_o$; it can be represented in terms of complete elliptic integrals:

$$Q(Z_o - Z'_o) = \frac{1}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\}$$

with

$$k^2 = \frac{4}{4 + (Z_o - Z'_o)^2}$$

A numerical method using discretization at half-integer points and using exact integration of the logarithmic singularity has been used. This method converts the problem of solving the integral equation to the problem of inverting a Toeplitz matrix. Results are presented and discussed.

* Information Resources Operation

† Mathematics Research Center, University of Wisconsin, Madison, WI

1. REPRESENTATION OF THE MAGNETIC FIELD

In this section we briefly outline the derivation of magnetic field representation in terms of spherical harmonics. Consider a single coil on a cylindrical surface, as shown in Figure 1. From the Biot-Savart Law we have:

$$\vec{B} = \frac{\mu_0}{4\pi} \int \vec{\nabla} \times \left(\frac{\vec{\lambda}}{R} \right) dA \quad (1)$$

$$R^2 = (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 \quad (2)$$

$$dA = a^2 d\phi_0 dZ_0$$

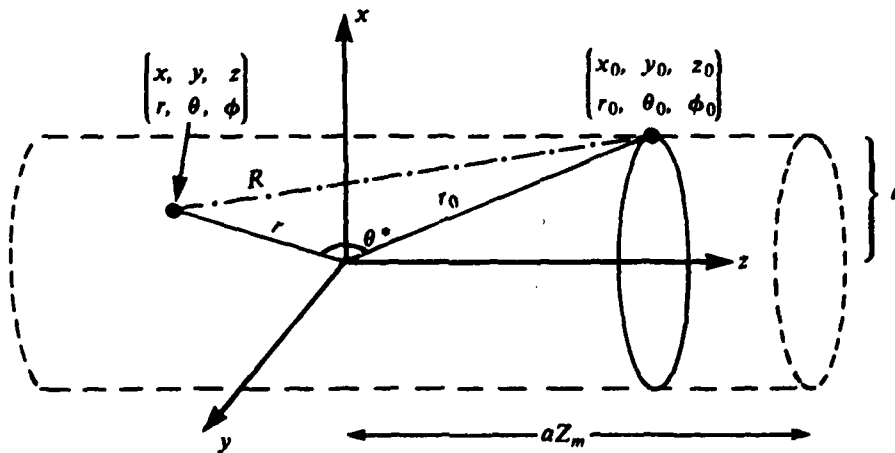


Figure 1.

where \vec{B} is the magnetic field, $\vec{\lambda}$ is the surface current-density vector and has the dimensions of amperes/meter. R is the distance between field (x, y, z) and source (x_0, y_0, z_0) points. The z -component of Eq. 1 can be represented in the differential form as:

$$dB_z = \frac{\mu_0 dA}{4\pi} \left\{ \lambda_x \frac{\partial}{\partial y_0} \left(\frac{1}{R} \right) - \lambda_y \frac{\partial}{\partial x_0} \left(\frac{1}{R} \right) \right\} \quad (3)$$

In Eq. 3 we have taken the derivatives with respect to source rather than field variables. Hence the change in sign. Using the notation of Figure 1, R may be expanded in terms of Legendre polynomials (Ref. 1, p. 173):

$$\frac{1}{R} = \frac{1}{(r^2 + r_0^2 - 2rr_0 \cos \theta^*)^{1/2}} = \begin{cases} \frac{1}{r_0} \sum_{n=0}^{\infty} \left(\frac{r}{r_0} \right)^n P_n(\cos \theta^*), & r < r_0 \\ \frac{1}{r} \sum_{n=0}^{\infty} \left(\frac{r_0}{r} \right)^n P_n(\cos \theta^*), & r > r_0 \end{cases} \quad (4)$$

In Eq. 4 the argument of the Legendre polynomials is θ^* , the angle between the field and source points. It is useful to use the biaxial harmonic expansion (Ref. 2, p. 164) to write

these functions in terms of (θ, ϕ) and (θ_0, ϕ_0) , the angles in the spherical coordinate representation of the field and source points.

$$P_n(\cos \theta) = \sum_{m=0}^{n-n} (2-\delta_m^0) \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta_0) P_n^m(\cos \theta) \cos m(\phi - \phi_0) \quad (5)$$

Here δ_m^0 is the Kronecker delta. It is equal to one when $m=0$ and is equal to zero otherwise. Substituting Eq. 5 into Eq. 4, we have:

$$\frac{1}{R} = \begin{cases} \frac{1}{r_0} \sum_{n=0}^{\infty} \sum_{m=0}^{n-n} \left(\frac{r}{r_0} \right)^n (2-\delta_m^0) \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) P_n^m(\cos \theta_0) \cos m(\phi - \phi_0), & r < r_0 \\ \frac{1}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n-n} \left(\frac{r_0}{r} \right)^n (2-\delta_m^0) \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) P_n^m(\cos \theta_0) \cos m(\phi - \phi_0), & r > r_0 \end{cases} \quad (6)$$

It can be seen from Eq. 3 that we need the cartesian derivatives of this representation. The necessary formulas can be derived from the integral representation of the solid spherical harmonics (Ref. 4, p. 1270). These harmonic functions are given by (Ref. 3, p. 369):

$$\begin{aligned} r^n C_{nm} &= r^n P_n^m(\cos \theta) \cos m\phi \\ r^n S_{nm} &= r^n P_n^m(\cos \theta) \sin m\phi \\ r^{-n-1} C_{nm} &= r^{-n-1} P_n^m(\cos \theta) \cos m\phi \\ r^{-n-1} S_{nm} &= r^{-n-1} P_n^m(\cos \theta) \sin m\phi \end{aligned} \quad (7)$$

Using the integral representation given in Ref. 4 (p. 1270), and after some algebraic manipulation, it can be shown that:

$$\begin{aligned} \frac{\partial}{\partial x} (r^{-n-1} C_{nm}) &= (1+\delta_m^0) \left\{ -\frac{1}{2} r^{-n-2} C_{n+1, m+1} + \frac{1}{2} (n-m+1)(n-m+2) r^{-n-2} C_{n+1, m-1} \right\} \\ \frac{\partial}{\partial y} (r^{-n-1} C_{nm}) &= (1+\delta_m^0) \left\{ -\frac{1}{2} r^{-n-2} S_{n+1, m+1} - \frac{1}{2} (n-m+2)(n-m+1) r^{-n-2} S_{n+1, m-1} \right\} \end{aligned} \quad (8)$$

Additional expressions for the cartesian derivatives of these functions are given in Ref. 5. Substituting Eq. 8 into Eq. 3 we have:

$$dB_z = - \frac{\mu_0 dA}{4\pi} \sum_{n=0}^{\infty} \sum_{m=0}^{n-1} r^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta) \\ \left[\lambda_y r_0^{-n-2} \left\{ -C_{n+1,m+1} + (n-m+1)(n-m+2) C_{n+1,m-1} \right\} \cos m\phi \right. \\ \left. + \lambda_x r_0^{-n-2} \left\{ S_{n+1,m+1} + (n-m+2)(n-m+1) S_{n+1,m-1} \right\} \sin m\phi \right] \quad (9)$$

Similar expansions can be derived for the other cartesian components of the magnetic field, and for $r > r_0$. Apparently these expansions have not been previously published.

Equation 9 represents an expansion of the z-component of the incremental magnetic field produced by an arbitrary element of surface current. It will converge for any origin such that $r < r_0$. Each term is the product of a factor ($r^n P_n^m(\cos \theta) \cos m\phi$ or $r^n P_n^m(\cos \theta) \sin m\phi$) that depends only on the field coordinates and of a factor that depends only on the source coordinates.

2. THE CLASSICAL APPROACH TO COIL DESIGN

For illustrative purposes consider a cylindrically symmetric case, with a circular surface current density which is a function only of Z_o

$$\lambda_x = -c \sigma_\phi(Z_o) \sin \phi_o \quad \lambda_\phi = c \sigma_\phi(Z_o) \quad (10)$$

$$\lambda_y = c \sigma_\phi(Z_o) \cos \phi_o \quad (11)$$

The function $\sigma_\phi(Z_o)$ is a dimensionless "shape function". The constant c is determined from the total number of ampere-turns, $N_1 I$, on the coil and by the normalization of $\sigma_\phi(Z_o)$.

$$N_1 I = \int_{-Z_m}^{Z_m} |\lambda_\phi| dZ_o = c a w_a \quad c = \frac{N_1 I}{a w_a} \quad w_a = \int_{-Z_m}^{Z_m} |\sigma_\phi(Z_o)| dZ_o$$

With $dA = a^2 d\phi_o dZ_o$, integration of Eq. 9 over the variable ϕ_o gives zero for those terms with $m \neq 0$. We are left with:

$$B_z = \frac{\mu_o c}{2} \sum_{n=0}^{\infty} \left\{ \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) \sum_{n=0}^{\infty} \frac{P_{n+1}^1(\cos \theta_o)}{r_o^{n+2}} dZ_o \right\} r^n P_n(\cos \theta) \quad (12)$$

Substituting $r_o^2/a^2 = (1+Z_o^2)$, we have

$$B_z = \sum_{n=0}^{\infty} A_n r^n P_n(\cos \theta) \quad (13)$$

$$A_n = \frac{\mu_o c}{2a^n} \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) \frac{P_{n+1}^1(\cos \theta_o)}{(1+Z_o^2)^{\frac{n+2}{2}}} dZ_o = \frac{\mu_o c \gamma_n}{2a^n} \quad (14)$$

Now, consider a pair of discrete coils which can be represented by

$$\sigma_\phi(Z_o) = \delta(Z_o - Z_c) + \delta(Z_o + Z_c). \quad (15)$$

We are not, at the moment, concerned about the normalization of $\sigma_\phi(Z_o)$. From Eq. 14 we have

$$A_n = \frac{\mu_o}{2a^n (1+Z_c^2)^{\frac{n+2}{2}}} \left\{ P_{n+1}^1(\cos \theta_c) + P_{n+1}^1(-\cos \theta_c) \right\} \quad n=0,2,4, \dots \quad (16)$$

$$A_n = 0$$

$$n=1,3,5, \dots$$

A first attempt at achieving a uniform field is to pick Z_c such that the coefficient of the second order ($n=2$) term in the expansion vanishes. That is

$$P_3^1(\cos \theta_c) + P_3^1(-\cos \theta_c) = -3(\sin \theta_c) (5\cos^2 \theta_c - 1) = 0 \quad (17)$$

This gives $\theta_c = \cos^{-1}(\sqrt{1/5})$. This result implies that the coils should be separated by a distance of one radius from one another.

A gradient field can be produced by using a discrete pair with the current reversed in one coil. Then

$$\sigma_\phi(Z_o) = \delta(Z_o - Z_c) - \delta(Z_o + Z_c)$$

Using Eq. 14 we obtain

$$A_n = \frac{\mu_0 c}{2a^n(1+Z_c^2)^{\frac{n+2}{2}}} \left\{ P_{n+1}^1(\cos\theta_c) - P_{n+1}^1(-\cos\theta_c) \right\}, \quad n=1,3,5,\dots \quad (18)$$

$$A_n = 0 \quad n=0,2,4,\dots$$

To obtain a gradient coil corrected through the third term in the expansion we need

$$P_4^1(\cos\theta_c) - P_4^1(-\cos\theta_c) = 5(\sin\theta_c)(7\cos^3\theta_c - 3\cos\theta_c) = 0. \quad (19)$$

The useful root of Eq. 19 is given by $\theta_c = \cos^{-1}(\sqrt{3}/7)$. This implies that the coils should be separated by a distance of $\sqrt{3}$ times the radius of the coil. Note that these two solutions are given by the zeroes of the functions g_2 and g_3 in Appendix 1.

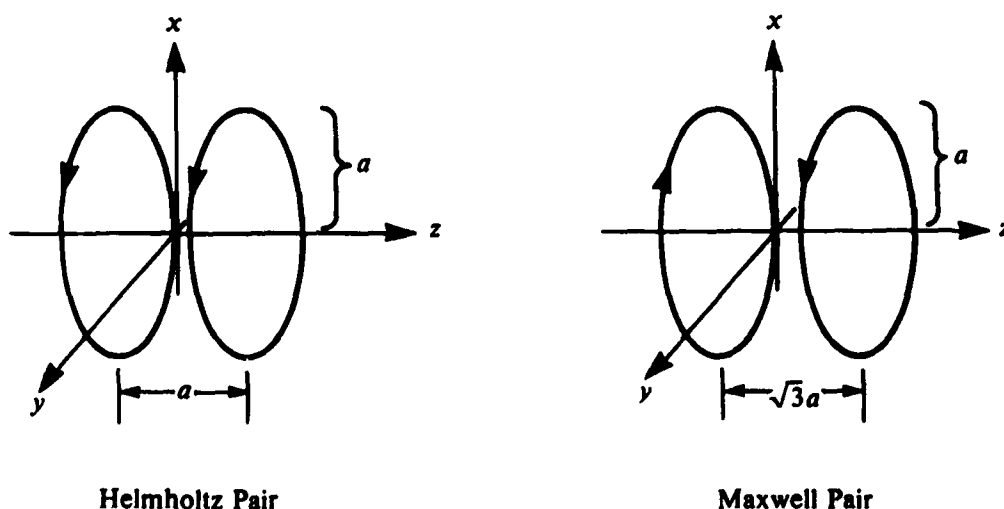


Figure 2.

The two coils discussed above are referred to respectively as the Helmholtz pair (used for uniform fields) and the Maxwell pair (used for linear gradients). Both these designs were discovered in the nineteenth century (Ref. 6, pp. 356-359).

In this century the technique outlined above has been refined by several authors (e.g., Ref. 8). The greatest advances have been accomplished by M.W. Garrett (Ref. 7), who provided algorithms for correcting the fields of cylindrically symmetric coil systems through arbitrarily high orders. He also described methods for designing such coils with arbitrary rectangular cross sections, thereby removing the restriction to discrete filaments.

The requirement that a given coefficient or coefficients in the expansion of Eq. 13 be zero does not, however, uniquely specify a coil design. In fact, it is easy to see that, if we permit the use of multiple pairs of coils, there are an infinite number of designs that will, for example, make the coefficient A_2 equal to zero. In the next section we outline a method for choosing among the large set of coils that meet a given requirement on the expansion coefficients.

3. THE DERIVATION OF THE INTEGRAL EQUATION

A common design problem is to find a coil that will produce a field that meets certain homogeneity criteria over a prescribed region of space. This requirement is usually given in terms of the expansion coefficients using an origin at the center of the region of interest. However, the coil will necessarily create fields in regions of space other than the region of interest. It is often desirable to keep the energy in this unused region to a minimum. This is particularly true when it is necessary to pulse the field on and off rapidly. Therefore we have investigated the problem of designing coils that will meet certain prescribed field characteristics and, at the same time, minimize the total stored energy in the magnetic field. In free space each cartesian component of \vec{B} can be shown to satisfy Laplace's equation and, therefore, can be written as a sum of solid spherical harmonics (Ref. 4, p. 1271). Although the method can be applied to more general situations, because of their technical and historical importance, we will consider, for the rest of this paper, coils wound on the surface of a cylinder and required to produce a z -component, B_z , of a prescribed character. The energy (in terms of inductance, L , and current, I) is given by (Ref. 2, p. 332):

$$W = \frac{1}{2} L I^2 = \frac{1}{2} \int \vec{\lambda} \cdot \vec{A} dA \quad (20)$$

where \vec{A} is the vector potential. From Poisson's solution to Laplace's equation (Ref. 1, pp. 167, 230)

$$\vec{A} = \frac{\mu_0}{4\pi} \int \frac{1}{R} \vec{\lambda} dA \quad (21)$$

Consider a circular surface current with cylindrical symmetry

$$\vec{\lambda} = c \sigma_\phi(Z_0) \{-\sin \phi_0 \hat{i} + \cos \phi_0 \hat{j}\} = c \sigma_\phi(Z_0) \hat{\phi} \quad (22)$$

Substituting from Eqs. 22 and 21 into Eq. 20, and carrying out two integrations over angular variables (see Ref. 2, p. 290 and Appendix 2), we have (using the notation of the previous sections)

$$\frac{1}{2} L I^2 = c^2 \mu_0 a^3 \int_{-Z_m}^{Z_m} \int_{-Z_m}^{Z_m} \frac{\sigma_\phi(Z_0) \sigma_\phi(Z'_0)}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\} dZ_0 dZ'_0 \quad (23)$$

where $K(k)$ and $E(k)$ are the complete elliptic integrals of the first and second kind, respectively, and

$$k^2 = \frac{4}{4 + (Z_0 - Z'_0)^2} \quad (24)$$

The magnetic field along the axis is given by Eqs. 13 and 14 as:

$$B_z = \sum_{n=0}^{\infty} A_n r^n P_n(\cos \theta) \quad (25)$$

with

$$A_n = \frac{\mu_0 c}{2a^n} \int_{-Z_m}^{Z_m} \sigma_\phi(Z_0) \frac{P_{n+1}^1(\cos \theta_0) dZ_0}{(1 + Z_0^2)^{\frac{n+2}{n}}} \quad (26)$$

Using Lagrange multipliers, we construct the functional as

$$I(\sigma_\phi) = \int_{-Z_m}^{Z_m} \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) \sigma_\phi(Z'_o) Q(Z_o - Z'_o) dZ_o dZ'_o + \sum_n \lambda_n \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) f_n(Z_o) dZ_o \quad (27)$$

with the kernel

$$Q(Z_o - Z'_o) = \frac{1}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\} \quad (28)$$

and

$$f_n(Z_o) = \frac{P_{n+1}^1(\cos \theta_o)}{(1+Z_o^2)^{\frac{n+2}{2}}} = \frac{g_n(Z_o)}{(1+Z_o^2)^{\frac{2n+3}{2}}} \quad (29)$$

The $g_n(Z_o)$ are polynomials that can be used to express the associated Legendre functions in cylindrical coordinates (Ref. 2, p. 215). They are listed in Appendix 1. To make Eq. 27 stationary near the exact solution, $\sigma_\phi(Z_o)$, let the trial function, $\sigma_\phi^*(Z_o)$, vary in the neighborhood of the exact solution, i.e.,

$$\sigma_\phi^*(Z_o) = \sigma_\phi(Z_o) + \epsilon \eta(Z_o) \quad (30)$$

where ϵ is a small parameter and $\eta(Z_o)$ is an arbitrary function. Substituting Eq. 30 into Eq. 27, we have

$$I(\sigma_\phi^*) = I(\sigma_\phi) + \epsilon \int_{-Z_m}^{Z_m} \eta(Z'_o) \left\{ \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) Q(Z_o - Z'_o) dZ_o + \sum_n \lambda_n f_n(Z'_o) \right\} dZ'_o + O(\epsilon^2) \quad (31)$$

Equation 31 will be stationary around the exact solution if the coefficient of ϵ vanishes, i.e.,

$$\int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) Q(Z_o - Z'_o) dZ_o + \sum_n \lambda_n f_n(Z'_o) = 0 \quad (32)$$

Equation 32 is a linear Fredholm integral equation of the first kind. It is to be solved for the unknown function, $\sigma_\phi(Z_o)$, which will minimize the energy. The λ_n are to be determined from the prescribed constraints. For example, if we wish to create a uniform field with zero coefficients for the quadratic and quartic errors, three constraints are required

$$\begin{aligned} \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) f_0(Z_o) dZ_o &= 1 \\ \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) f_2(Z_o) dZ_o &= 0 \quad (2^{\text{nd}} \text{ order correction}) \\ \int_{-Z_m}^{Z_m} \sigma_\phi(Z_o) f_4(Z_o) dZ_o &= 0 \quad (4^{\text{th}} \text{ order correction}). \end{aligned} \quad (33)$$

The first of these constraints serves to normalize $\sigma_\phi(Z_o)$. An arbitrarily high degree of homogeneity can be achieved by requiring any desired number of additional coefficients, A_n , to be zero. It is noted that the kernel has a logarithmic singularity at $Z_o = Z'_o$, i.e., by using

the expansions for $K(k)$ and $E(k)$ near $k=1$ (Ref. 11, p. 73), we have

$$Q(Z) = \frac{1}{2} (U-2) + \frac{Z^2}{32} (3U-1) - \frac{Z^4}{4096} [30U-31] \\ + \frac{Z^6}{196608} [210U-247] + O(Z^8) \quad (34)$$

$$U = \ln \left[\frac{8}{|Z|} \right] \quad Z \ll 1$$

It has not been possible to obtain the solution of the integral equation in closed form. However, an approximate numerical solution may be found by discretizing Eq. 32 (Ref. 13). We divide the region from $-Z_m$ to Z_m into $2N$ subunits of width $\delta Z_0 = Z_m/N$. We construct an array of linear equations

$$\sum_{j=1}^{2N} A_{ij} \sigma_j = f_i^n \quad (33')$$

where

$$f_i^n = f_n \left(\frac{Z_m}{N} (i - 1/2 - N) \right)$$

Here σ_j and f_i are the values of the associated functions at the half-integer points and

$$A_{ij} = \begin{cases} \frac{Z_m}{N} Q \left(\frac{Z_m}{N} |i-j| \right) & i \neq j \\ \frac{Z_m}{2N} \left(\ln \left[\frac{16N}{Z_m} \right] - 1 \right) & i = j \end{cases}$$

The desired solution is found by the inversion of the matrix A_{ij} . We have used the IMSL routine LEQT1F (Ref. 12). The linearity of the system permits us to solve Eq. 33' for an arbitrary $f_n(Z_0)$, and then to use superposition to meet the constraints and determine the Lagrange multipliers. We thereby produce an approximate numerical solution to the system of equations 32 and 33.

The matrix A_{ij} is seen to have the Toeplitz property. That is A_{ij} is a function of $|i-j|$ only. Because $Q(Z)$ is singular at $Z=0$, A_{ij} for $i=j$ is computed by integrating the first term in the expansion in Eq. 34 across the appropriate interval.

$$A_{jj} = \int_{-Z_m/2N}^{Z_m/2N} \left\{ \frac{1}{2} \ln \left[\frac{8}{|Z|} \right] - 1 \right\} dZ$$

We have used this method to solve for the $\sigma_\phi(Z_0)$ that minimize the total energy and provide for field uniformity by eliminating coefficients through the 18th order for various values of Z_m . We have also applied the method to the design of gradient coils required to achieve a high order of linearity by eliminating coefficients through the ninth order. These results will be given in a forthcoming report.

Figures 3 and 4 illustrate the solutions obtained for zero order and second order constraints on the homogeneity, respectively. In these examples $Z_m=3$ and $N=90$. Note that the solution with no homogeneity constraint can be approximated by a single short, uniform solenoid centered on $Z_0=0$. Also the solution where the second order term is zero can be approximated by a pair of short solenoids approximately centered at the location of the Helmholtz pair. In general it appears that although $\sigma_\phi(Z_0)$, the solution that actually mini-

mizes the stored energy, will vary continuously with position, there will always be designs with only slightly increased stored energy that are a series of short uniform solenoids centered at the maxima of $\sigma_\phi(Z_o)$.

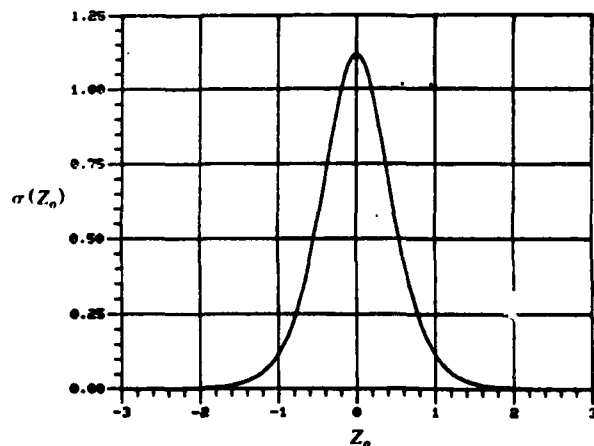


Figure 3. No corrections.

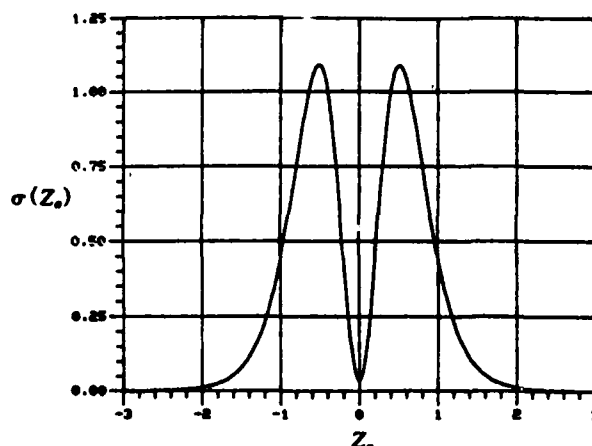


Figure 4. Second order correction.

The solutions obtained for the values of Z_o not close to $\pm Z_m$ have been sufficiently accurate for practical purposes for $N \geq 50$. Increasing N from 50 to 90 changed these values only by a few percent. However, the solutions contain a singularity as Z_o approaches Z_m . This singularity is sufficiently weak for large Z_m that it is not apparent in Figures 3 and 4. The singularity is apparent in all solutions obtained for $Z_m \leq 1.5$ or so, and is discussed in the next section.

4. ANALYSIS OF THE EDGE SINGULARITY

The logarithmic singularity in the kernel appears to force the solution to have a singularity of the form

$$\frac{C}{\sqrt{Z_m^2 - Z_o^2}} \quad (34')$$

as Z_o approaches $\pm Z_m$ (Refs. 10; 16, p. 536ff; 17, p. 447ff). Here C is a constant determined by the details of the problem. This integrable singularity is sufficiently weak that it ordinarily will have no practical effect on coil design. However, it places in question the validity of approximating $\sigma_\phi(Z_o)$ by its value at the half integer points for the points nearest the edges at $Z_o = \pm Z_m$.

This edge effect can be investigated by using the discretization process developed in the previous section to solve an analogous integral equation which has a known closed form solution (Ref. 9, p. 143). Consider the integral equation

$$\int_{-Z_m}^{Z_m} \ln |t-s| f(s) ds = -\pi \ln \left[\frac{2}{Z_m} \right] \quad -Z_m < t < Z_m. \quad (35)$$

This equation resembles Eq. 32 in that it is a linear Fredholm equation of the first kind and the kernel has a logarithmic singularity at $s=t$. However, the exact solution to Eq. 35 is known to be

$$f(s) = \frac{1}{(Z_m^2 - s^2)^{1/2}}. \quad (36)$$

We convert, as before, from integral equation (35) to a system of linear algebraic equations

$$\sum_{j=1}^{2N} A'_{ij} f_j = -\pi \ln \left[\frac{2}{Z_m} \right] \quad (37)$$

by taking

$$f_j = f \left(\frac{Z_m}{N} \left[j - N - \frac{1}{2} \right] \right) \\ A'_{ij} = \begin{cases} \frac{Z_m}{N} \ln \left[\frac{Z_m}{N} |i-j| \right] & i \neq j \\ \frac{Z_m}{N} \left[\ln \left(\frac{Z_m}{N} \right) - 1 \right] & i=j \end{cases} \quad (38)$$

for $i=j$, A'_{ij} is computed, as before, by integrating across the singularity.

$$A'_{ij} = \int_{-Z_m/2N}^{Z_m/2N} \ln |s| ds$$

Consider f_1 and f_{2N} , the approximations to $f(s)$ at the extreme edge positions obtained by inverting the Toeplitz matrix in Eq. 37. By symmetry $f_1 = f_{2N}$. The exact value of $f(s)$ at midpoint of the edge interval is

$$f^* = \frac{1}{\sqrt{Z_m^2 - s_{2N}^2}}$$

where $s_{2N} = Z_m \left(1 - \frac{1}{2N} \right)$

Table I illustrates, for several values of N (for $Z_m = 1$), the ratio of f_{2N}/f^* and the associated correction factor discussed below. The third column of Table I shows that, despite the presence of the edge singularity, the numerical method provides an estimate which is only about 33 percent too high. This estimate can be further improved in a second approximation by dividing f_1 and f_{2N} by a correction factor, c_f

$$c_f = \frac{\sqrt{4N-1} \left(\sqrt{2} \ln \left(\frac{Z_m}{2N} \right) + D \right)}{2\sqrt{N} \left(\ln \left(\frac{Z_m}{2N} \right) - 1 \right)} \quad (39)$$

$$D = \ln \left(\frac{\sqrt{2}+1}{\sqrt{2}-1} \right) - 2\sqrt{2} = -1.0657$$

This factor is obtained by integrating exactly the known form of the solution (containing the square root singularity) multiplied by the kernel (containing the logarithmic singularity) over the intervals at the edges. It does not depend on the value of C in Eq. 34'.

Table I
EDGE EFFECTS FOR $Z_m = 1$
AND SEVERAL VALUES OF N

N	c_f	f_{2N}/f^*
10	1.3103	1.3161
20	1.3315	1.3276
30	1.3402	1.3314
40	1.3452	1.333
50	1.3486	1.3341
∞	$\sqrt{2}$	—

The important feature of the correction factor in Eq. 39 is that it is independent of the details of the numerical solution and depends only on the form of the singularity at the edges. The apparent convergence of the two columns in Table I as N increases suggests that the error in the values of f_1 and f_{2N} can be substantially improved by simply dividing them by the factor c_f without further modification of the matrix equation.

Although we have not investigated the matter thoroughly, the results described above indicate that the presence of the edge singularity does not appreciably affect the accuracy of the method for interior points. Furthermore, if it is desired to obtain a more accurate value for the points at the edges (ordinarily this is not of practical importance), the simple procedure outlined above is available.

CONCLUSIONS

We have shown that a method is available for designing coils that meet prescribed homogeneity requirements and that minimize a particular figure of merit, the stored energy. This permits us to make a rational choice between the infinite number of possible coils that

will meet the homogeneity requirements if no constraints are applied. Interestingly, although the solutions vary continuously with Z_0 , they can be closely approximated by discrete uniform coils which have only slightly greater values of stored energy and are much easier to build. The method has been illustrated for two-dimensional windings on the surface of a cylinder because of the great technical importance of this case. However, it is probable that the method can be readily generalized to other geometries and to different figures of merit.

ACKNOWLEDGMENT

It is a pleasure to acknowledge the help of Maria Barnum in preparing this difficult manuscript for publication.

REFERENCES

1. J.A. Stratton, *Electromagnetic Theory*, McGraw-Hill Book Company, New York, NY, 1941.
2. W.R. Smythe, *Static and Dynamic Electricity*, Third Edition, McGraw-Hill Book Company, New York, NY, 1968.
3. W.D. MacMillan, *The Theory of the Potential*, Dover Publications, Inc., New York, NY, 1958.
4. P.M. Morse and H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill Book Company, New York, NY, 1953.
5. J.F. Schenck and M.A. Hussain, "Formulation of Design Rules for NMR Imaging Coils by Symbolic Manipulation," *Proceedings of the 1981 ACM Symposium on Symbolic and Algebraic Computation* (P.S. Wang, ed.), 85-93.
6. J.C. Maxwell, *A Treatise on Electricity and Magnetism*, Dover Publications Inc., New York, NY, 1954.
7. M.W. Garrett, "Thick Cylindrical Coil Systems for Strong Magnetic Fields with Field or Gradient Homogeneities of the 6th to 20th Order," *Journal of Applied Physics* 38, 2563-2586 (1967).
8. L.W. McKeehan, "Combination of Circular Currents for Producing Uniform Magnetic Fields," *Review of Scientific Instruments*, 150-153 (1936).
9. W. Magnus and F. Oberhettinger, *Formulas and Theorems for the Functions of Mathematical Physics*, Chelsea Publishing Company, New York, NY, 1954.
10. B. Noble and M. Hussain, "A Variational Method for Inclusion and Indentation Problems," *The Journal of the Institute of Mathematics and Its Applications* 5, 194-205 (1969).
11. E. Jahnke and F. Emde, *Tables of Functions with Formulae and Curves*, Dover Publications, Inc., New York, NY, 1945.
12. *IMSL Library Reference Manual*, Edition 8, IMSL, 7500 Bellaire Blvd., Houston, TX, 1980.
13. M.A. Jaswon and G.T. Symm, *Integral Equation Methods in Potential Theory and Elastostatics*, Academic Press, New York, NY, 1978.
14. I.S. Gradshteyn and I.M. Ryzhik, *Tables of Integrals, Series and Products, Corrected and Enlarged Edition*, (prepared by A. Jeffrey) Academic Press, New York, NY, 1980.
15. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. 1, Interscience Publishers, Inc., New York, NY, 1953.
16. F.D. Gakhov, *Boundary Value Problems*, (translation edited by I.N. Sneddon) Pergamon Press, London, 1966.
17. N.I. Muskhelishvili, *Some Basic Problems in the Theory of Elasticity* (translated by J.R.M. Radok), P. Noordhoff, Groningen-The Netherlands, 1963.

Appendix 1

TABULATION OF THE FUNCTIONS $g_n(Z)$

$$g_0 = 1$$

$$g_1 = 3Z$$

$$g_2 = \frac{3}{2} (2Z - 1)(2Z + 1)$$

$$g_3 = \frac{5Z}{2} (4Z^2 - 3)$$

$$g_4 = \frac{15}{8} (8Z^4 - 12Z^2 + 1)$$

$$g_5 = \frac{21Z}{8} (8Z^4 - 20Z^2 + 5)$$

$$g_6 = \frac{7}{16} (64Z^6 - 240Z^4 + 120Z^2 - 5)$$

$$g_7 = \frac{9Z}{16} (64Z^6 - 336Z^4 + 280Z^2 - 35)$$

$$g_8 = \frac{45}{128} (128Z^8 - 896Z^6 + 1120Z^4 - 280Z^2 + 7)$$

$$g_9 = \frac{55Z}{128} (128Z^8 - 1152Z^6 + 2016Z^4 - 840Z^2 + 63)$$

$$g_{10} = \frac{33}{256} (512Z^{10} - 5760Z^8 + 13440Z^6 - 8400Z^4 + 1260Z^2 - 21)$$

$$g_{11} = \frac{39Z}{256} (512Z^{10} - 7040Z^8 + 21120Z^6 - 18480Z^4 + 4620Z^2 - 231)$$

$$g_{12} = \frac{91}{1024} (1024Z^{12} - 16896Z^{10} + 63360Z^8 - 73920Z^6 + 27720Z^4 - 2772Z^2 + 33)$$

$$g_{13} = \frac{105Z}{1024} (1024Z^{12} - 19968Z^{10} + 91520Z^8 - 137280Z^6 + 72072Z^4 - 12012Z^2 + 429)$$

Appendix 1 (Cont'd)

$$g_{14} = \frac{15}{2048} (16384Z^{14} - 372736Z^{12} + 2050048Z^{10} - 3843840Z^8 + 2690688Z^6 \\ - 672672Z^4 + 48048Z^2 - 429)$$

$$g_{15} = \frac{17Z}{2048} (16384Z^{14} - 430080Z^{12} + 2795520Z^{10} - 6406400Z^8 + 5765760Z^6 \\ - 2018016Z^4 + 240240Z^2 - 6435)$$

$$g_{16} = \frac{153}{32768} (32768Z^{16} - 983040Z^{14} + 7454720Z^{12} - 20500480Z^{10} + 23063040Z^8 \\ - 10762752Z^6 + 1921920Z^4 - 102960Z^2 + 715)$$

$$g_{17} = \frac{171Z}{32768} (32768Z^{16} - 1114112Z^{14} + 9748480Z^{12} - 31682560Z^{10} + 43563520Z^8 \\ - 26138112Z^6 + 6534528Z^4 - 583440Z^2 + 12155)$$

$$g_{18} = \frac{95}{65536} (131072Z^{18} - 5013504Z^{16} + 50135040Z^{14} - 190095360Z^{12} + 313657344Z^{10} \\ - 235243008Z^8 + 78414336Z^6 - 10501920Z^4 + 437580Z^2 - 2431)$$

$$g_{19} = \frac{105Z}{65536} (131072Z^{18} - 5603328Z^{16} + 63504384Z^{14} - 277831680Z^{12} + 541771776Z^{10} \\ - 496624128Z^8 + 212838912Z^6 - 39907296Z^4 + 2771340Z^2 - 46189)$$

$$g_{20} = \frac{231}{262144} (262144Z^{20} - 12451840Z^{18} + 158760960Z^{16} - 793804800Z^{14} + 1805905920Z^{12} \\ - 1986496512Z^{10} + 1064194560Z^8 - 266048640Z^6 + 27713400Z^4 \\ - 923780Z^2 + 4199)$$

Appendix 2

DERIVATION OF THE STORED ENERGY AS A QUADRATIC INTEGRAL FORM

Equation 27 can be derived from Eq. 20 as follows.

$$W = \frac{\mu_0}{8\pi} \iint_{\substack{\text{cylindrical} \\ \text{surface twice}}} \frac{\bar{\lambda} \cdot \bar{\lambda}'}{R} dA dA'$$

$$\bar{\lambda} = c \sigma_\phi(Z_0) \hat{\phi} \quad \bar{\lambda}' = c \sigma_\phi(Z'_0) \hat{\phi}'$$

$$\hat{\phi} = -\sin\phi \hat{i} + \cos\phi \hat{j} \quad Z_0 = z_0/a$$

$$\hat{\phi} \cdot \hat{\phi}' = \sin\phi_0 \sin\phi'_0 + \cos\phi_0 \cos\phi'_0 = \cos(\phi'_0 - \phi_0)$$

$$dA = a^2 d\phi_0 dZ_0 \quad dA' = a^2 d\phi'_0 dZ'_0$$

$$\frac{1}{R} = \frac{1}{[2a^2 - 2a^2 \cos(\phi'_0 - \phi_0) + (z_0 - z'_0)^2]^{1/2}}$$

or,

$$\frac{1}{R} = \frac{1}{a} \frac{1}{[2 - 2\cos(\phi'_0 - \phi_0) + (Z_0 - Z'_0)^2]^{1/2}}$$

$$W = \frac{\mu_0 c^2 a^3}{8\pi} \int_{-Z_m}^{Z_m} \int_{-Z_m}^{Z_m} \int_0^{2\pi} \int_0^{2\pi} \frac{\cos(\phi'_0 - \phi_0) \sigma_\phi(Z_0) \sigma_\phi(Z'_0)}{[2 - 2\cos(\phi'_0 - \phi_0) + (Z'_0 - Z_0)^2]^{1/2}} d\phi'_0 d\phi_0 dZ'_0 dZ_0$$

$$\text{Let } I_1 = \int_0^{2\pi} \frac{\cos(\phi'_0 - \phi_0)}{[2 - 2\cos(\phi'_0 - \phi_0) + (Z'_0 - Z_0)^2]^{1/2}} d\phi'_0$$

with ϕ_0, Z_0, Z'_0 held constant, substitute $\phi''_0 = \phi'_0 - \phi_0$

$$I_1 = \int_{-\phi_0}^{2\pi - \phi_0} \frac{\cos\phi''_0 d\phi''_0}{[2 - 2\cos\phi''_0 + (Z_0 - Z'_0)^2]^{1/2}}$$

$$\text{Let } \phi''_0 = 2\psi + \pi$$

$$d\phi''_0 = 2d\psi$$

$$\cos\phi''_0 = 2\sin^2\psi - 1$$

$$I_1 = 2 \int_{-\frac{\phi_0}{2} - \frac{\pi}{2}}^{\frac{-\phi_0}{2} + \frac{\pi}{2}} \frac{(2\sin^2\psi - 1) d\psi}{[4 - 4\sin^2\psi + (Z_0 - Z'_0)^2]^{1/2}}$$

$$\text{let } k^2 = \frac{4}{4 + (Z_0 - Z'_0)^2}$$

and note that, because $\sin^2\psi$ is periodic with a period equal to π ,

$$\int_A^{A+\pi} f(\sin^2\psi) d\psi = 2 \int_0^{\pi/2} f(\sin^2\psi) d\psi$$

and is independent of A for any function, f .

$$I_1 = k \int_0^\pi \frac{(2\sin^2\psi - 1)}{(1 - k^2\sin^2\psi)^{1/2}} d\psi$$

These integrals can be expressed in terms of $K(k)$ and $E(k)$, the complete elliptic integrals of the first and second kinds (Ref. 11, p. 162).

$$\int_0^{\pi/2} \frac{2\sin^2\psi}{(1 - k^2\sin^2\psi)^{1/2}} d\psi = 2 \left[\frac{1}{k^2} [K(k) - E(k)] \right]$$

$$\int_0^{\pi/2} \frac{1}{(1 - k^2\sin^2\psi)^{1/2}} d\psi = K(k)$$

Therefore,

$$I_1 = \frac{4}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\}$$

and

$$W = \frac{\mu_o c^2 a^3}{8\pi} \int_{-Z_m}^{Z_m} \int_{-Z_m}^{Z_m} \int_0^{2\pi} \frac{4}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\} \sigma(Z_o) \sigma(Z'_o) d\phi_o dZ'_o dZ_o.$$

The integrand is independent of ϕ_o , therefore,

$$W = \mu_o c^2 a^3 \int_{-Z_m}^{Z_m} \int_{-Z_m}^{Z_m} \frac{1}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\} \sigma(Z_o) \sigma(Z'_o) dZ'_o dZ_o.$$

Define S by

$$S = \int_{-Z_m}^{Z_m} \int_{-Z_m}^{Z_m} \frac{1}{k} \left\{ \left(1 - \frac{k^2}{2} \right) K(k) - E(k) \right\} \sigma(Z_o) \sigma(Z'_o) dZ'_o dZ_o.$$

This yields,

$$W = \mu_o c^2 a^3 S.$$

Using $c = \frac{N_l l}{aw_a}$ and $W = \frac{1}{2} L I^2$ we have

$$W = \frac{\mu_o N_l^2 I^2 a}{w_a^2} S \text{ and } L = \frac{2\mu_o N_l^2 a}{w_a^2} S.$$

From Eq. 14 we may also take $c = \frac{2a''}{\mu_o} \frac{A_n}{\gamma_n}$ yielding

$$W = \frac{4a^{2n+3}}{\mu_o} A_n^2 \frac{S}{\gamma_n^2}.$$

Note that S is a quadratic integral form in the sense discussed by Courant and Hilbert (Ref. 15, p. 122).

ELEMENT TYPE COMPARISON IN BASIN OSCILLATION ANALYSIS

Mark D. Prater
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

Keith W. Bedford
Department of Civil Engineering
The Ohio State University
Columbus, Ohio 43210

ABSTRACT. A comparison of finite element types in the solution of basin oscillations is presented. A two-dimensional finite model is used to obtain a harmonic solution to the linearized velocity potential formulation of the shallow water wave equation. The influence of three types of elements, linear triangles (LT), linear isoparametric quadrilaterals (LIQ), and quadratic isoparametric quadrilaterals (QIQ) on the computation of periods of oscillation and normalized displacement fields of basins is tested.

A brief description of the solution technique is given along with the results and conclusions of two model applications. The first employs a rectangular basin with uniform depth and the second a natural basin, Lake Erie, with varied geometry and bathymetry.

1. INTRODUCTION. Oscillations in enclosed basins are a phenomena which have been studied for many years. Analytical solutions for the partial differential equations which govern oscillatory behavior have been found for various mathematically defined basins. Numerical solutions have also been found by assuming that a natural basin can be approximated in a one-dimensional form, or that the depth is uniform throughout. Only since the late 1960's have approximate solutions been sought in basins of arbitrary geometry and bathymetry. This paper presents a simple, quick, and accurate method to determine periods of oscillations and normalized displacement fields for completely arbitrary enclosed basins.

2. DEVELOPMENT OF FINITE ELEMENT EQUATIONS. The equation governing oscillatory motion for a basin of arbitrary shape is

$$\frac{1}{g} \frac{\partial^2 \phi}{\partial t^2} = \frac{\partial}{\partial x} \left(h \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(h \frac{\partial \phi}{\partial y} \right) \quad (1)$$

where h is still-water depth and ϕ is velocity potential. Assuming a harmonic solution to remove time dependency, let

$$\phi(x, y, t) = \phi(x, y) e^{i\omega t}$$

where ω is the oscillation frequency.

Substituting into Equation 1 gives

$$-\frac{\omega^2 \phi}{g} = \frac{\partial}{\partial x} \left(h \frac{\partial \phi}{\partial y} \right) + \frac{\partial}{\partial y} \left(h \frac{\partial \phi}{\partial x} \right) \quad (2)$$

or in vector notation

$$\nabla \cdot (h \nabla \phi) = -\lambda \phi$$

where $\lambda = \omega^2/g$, which is the form for the eigenvalue problem. The equation essentially states that $[M]\{\phi\} = -\lambda\{\phi\}$ where $[M]$ contains information about the bathymetry and geometry of the basin. In solving this, the periods of oscillation can be determined from the eigenvalue λ by $1/T = [3600(g\lambda)^{1/2}]/2\pi$, and the corresponding displacement field from the eigenvector $\{\phi\}$ by elevation $\zeta = \phi(\lambda/g)^{1/2}$.

To discretize the equations over the domain of the problem, the finite element method is used, primarily for the flexibility in fitting the elements to irregular boundaries. The solution technique assumes that the unknown quantity ϕ , can be approximated by

$$\bar{\phi} = \sum_{i=1}^n N_i \phi_i = \langle N \rangle \{\phi\}$$

where ϕ_i = value of the variable at node i

n = number of nodes per element

N_i = value of the shape function at node i

Given a differential equation such that $L(\phi) = 0$ where L is a differential operator, and substitute the approximate variable, yields $L(\bar{\phi}) = R \neq 0$, where R is a residual value or error between the true and approximate solutions. One procedure used to reduce R to a minimum is to set $\int_A W_i R dA = 0$ where W_i is a weighting function evaluated at point i . This relationship means that the residual value is zero when averaged over the domain of the problem. Substituting for R gives $\int_A W_i L(\bar{\phi}) dA = 0$, or substituting further $\int_A W_i L(\langle N \rangle \{\phi\}) dA = 0$. The Galerkin Method of determining the weighting functions assumes that W_i is equal to N_i for all i . This gives the final formulation of $\int_A N_i L(\langle N \rangle \{\phi\}) dA = 0$. A similar procedure is done with our governing equation.

Expanding Equation 2 so that

$$h \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) + \frac{\partial h}{\partial x} \frac{\partial \phi}{\partial x} + \frac{\partial h}{\partial y} \frac{\partial \phi}{\partial y} = -\lambda \phi \quad (3)$$

and multiply through by N_i and integrate over the domain gives

$$\iint_A N_i \left[h \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) + \frac{\partial h}{\partial x} \frac{\partial \phi}{\partial x} + \frac{\partial h}{\partial y} \frac{\partial \phi}{\partial y} \right] dA = - \iint_A N_i \lambda \phi dA \quad (4)$$

To simplify the above, the first terms are separated and integrated part by part. Since

$$N_i h \frac{\partial^2 \phi}{\partial x^2} = \frac{\partial}{\partial x} \left(N_i h \frac{\partial \phi}{\partial x} \right) - h \frac{\partial N_i}{\partial x} \frac{\partial \phi}{\partial x} - N_i \frac{\partial h}{\partial x} \frac{\partial \phi}{\partial x} \quad (5)$$

integrating over the domain yields

$$\iint_A N_i h \frac{\partial^2 \phi}{\partial x^2} dx dy = \int_C N_i h \frac{\partial \phi}{\partial x} dy - \iint_A h \frac{\partial N_i}{\partial x} \frac{\partial \phi}{\partial x} dx dy - \iint_A N_i \frac{\partial h}{\partial x} \frac{\partial \phi}{\partial x} dx dy \quad (6)$$

where C denotes integration along the boundary. Doing the same for the y -direction terms and substituting back into Equation 4 gives

$$\iint_A h \left(\frac{\partial N_i}{\partial x} \frac{\partial \phi}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial \phi}{\partial y} \right) dx dy - \int_C N_i h \frac{\partial \phi}{\partial n} d\Gamma = \iint_A N_i \lambda \phi dx dy \quad (7)$$

where n is a direction normal, and Γ is a direction tangential to a boundary. For an enclosed basin with solid boundaries, the term $\partial \phi / \partial n$ is zero. The above then becomes

$$\iint_A h \left(\frac{\partial N_i}{\partial x} \frac{\partial \phi}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial \phi}{\partial y} \right) dx dy = \iint_A N_i \lambda \phi dx dy \quad (8)$$

3. ELEMENT DEVELOPMENT. Two types of elements are used to discretize the equation over the domain (Figure 1). The arbitrary shape of the element in the x - y global plane is transformed into a square in the local ξ - η plane. Since the elements cannot be integrated over analytically, the coordinate mapping is done for ease of numerical integration. The two coordinate systems are related by

$$\begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} = J \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} \quad (9)$$

where J is the Jacobian matrix. Using the shape function, the directional variables can be approximated by

$$x = \sum_{i=1}^n N_i x_i \quad \text{and} \quad y = \sum_{i=1}^n N_i y_i$$

Therefore,

$$J = \begin{bmatrix} \sum \frac{\partial(N_i x_i)}{\partial \xi} & \sum \frac{\partial(N_i y_i)}{\partial \xi} \\ \sum \frac{\partial(N_i x_i)}{\partial \eta} & \sum \frac{\partial(N_i y_i)}{\partial \eta} \end{bmatrix} \quad (10)$$

The global derivatives of the shape functions can be solved for, such that

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = J^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} \quad (11)$$

The inverse of the Jacobian is easily found by Cramer's rule.

The shape function N_i is defined as the function which is equal to one when evaluated at node i and is equal to zero at all other nodes of the element. Shape functions which satisfy this are

$$N_i = \frac{1}{4} (1 + \xi \xi_i)(1 + \eta \eta_i), \quad i = 1, 2, 3, 4$$

for the LIQ element and

$$N_i = \frac{1}{4} (1 + \xi \xi_i)(1 + \eta \eta_i)(\xi \xi_i + \eta \eta_i), \quad i = 1, 3, 5, 7$$

$$N_i = \frac{1}{2} (1 - \xi^2)(1 + \eta \eta_i), \quad \text{and} \quad i = 2, 6$$

$$N_i = \frac{1}{2} (1 + \xi \xi_i)(1 - \eta^2), \quad i = 4, 8$$

for the QIQ element where ξ and η are directional variables and ξ_i and η_i are coordinate values at node i .

To complete the transformation of coordinate systems, the determinant of the Jacobian matrix ($\det J$) is defined as

$$dx \, dy = \det J \, d\xi \, d\eta$$

This coordinate mapping makes the relation

$$\int_A f(x,y) dx dy = \int_{-1}^1 \int_{-1}^1 f(\xi,\eta) \det J d\xi d\eta \quad (12)$$

possible. The right-hand side of Equation 12 can now be solved by numerical integration.

The integration technique chosen in Gaussian Quadrature, since a $2K-1$ degree polynomial can be integrated exactly with only K points, thus reducing computation.

Gaussian Quadrature assumes that

$$\int_{-1}^1 \int_{-1}^1 f(\xi,\eta) d\xi d\eta = \sum_{i=1}^k \sum_{j=1}^k W_i W_j f(\xi_i, \eta_j)$$

where W_i and W_j are weighting factors at an integration point (i,j) within the element. Positions of these integration points are found in Figure 2 and Table 1.

Returning to Equation 8, and assuming that

$$h = \langle N_e \rangle \{h_e\} \quad \text{and} \quad \phi = \langle N_e \rangle \{\phi_e\},$$

where the subscript e denotes values for a specific element, and applying Gaussian Quadrature yields

$$\begin{aligned} & \left(\sum_{i=1}^k \sum_{j=1}^k W_i W_j \langle N_e \rangle \{h_e\} \right) \left\{ \left[J^{-1}(1,1) \frac{\partial \{N_e\}}{\partial \xi} + J^{-1}(1,2) \frac{\partial \{N_e\}}{\partial \eta} \right] \right. \\ & \times \left[J^{-1}(1,1) \frac{\partial \langle N_e \rangle}{\partial \xi} + J^{-1}(1,2) \frac{\partial \langle N_e \rangle}{\partial \eta} \right] + \left[J^{-1}(2,1) \frac{\partial \{N_e\}}{\partial \xi} \right. \\ & \left. \left. + J^{-1}(2,2) \frac{\partial \{N_e\}}{\partial \eta} \right] \left[J^{-1}(2,1) \frac{\partial \langle N_e \rangle}{\partial \xi} + J^{-1}(2,2) \frac{\partial \langle N_e \rangle}{\partial \eta} \right] \right\} \det J \{ \phi_e \} \\ & = \left[\sum_{i=1}^k \sum_{j=1}^k W_i W_j \left(\{N_e\} \langle N_e \rangle \det J \right) \right] \lambda \{ \phi_e \} \quad (13) \end{aligned}$$

In the above equation ϕ_e is no longer of function of space and is removed from the integration. Also, $J^{-1}(1,1)$ represents the value at location (1,1) of the inverted Jacobian matrix.

The equation can now be considered as

$$[S_e]\{\phi_e\} = \lambda[R_e]\{\phi_e\}$$

where S_e and R_e are n by n matrices and n is the number of nodes per element. After assembling the element matrices over the entire domain, the full equation becomes

$$[S]\{\phi\} = \lambda[R]\{\phi\}$$

where S and R are m by m matrices, where m is the number of nodes in the domain. Both S and R are symmetric, banded matrices, the bandwidth dependent on how the nodes were numbered. The eigenvalue solution scheme used here was discussed by Jennings, where iterations are carried out simultaneously with several trial vectors. Once the dominant eigenvalue and eigenvector have been found, they are removed from the set of trial vectors. This process of iteration and elimination is continued until all of the required values have been obtained. This process is quick and efficient when only a small number of the eigenvalues and vectors are needed.

4. TEST BASIN RESULTS. To effectively test the solution procedure, the governing equations are solved analytically for a rectangular basin with uniform bathymetry. The numerical solution is then calculated for the LIQ and QIQ element representations and compared to the analytical results to determine the correctness of the periods and displacement fields.

The analytical solution for the eigenvalues of the governing equation is given by Lamb as

$$\lambda = \pi^2 \left(\frac{m^2}{a^2} + \frac{n^2}{b^2} \right) \quad \begin{array}{l} m = 0, 1, 2, \dots \\ n = 0, 1, 2, \dots \end{array}$$

where a and b are the x and y dimensions of a unit depth rectangle, and m and n are the modes of oscillation in each directions. Values of $\sqrt{\lambda}$ are given in Table 2 for a basin of size 1.00 by 1.01 feet.

A previous study by Valizadeh-Alavi used a grid of 200 equally sized linear triangles (LT), shown in Figure 3. As compared in Table 2, the roots of the eigenvalues from the LT elements differ slightly from the analytical results, the error ranging up to over 4 percent. A study of the generated displacement fields show that for oscillations in one direction the LT elements give good results. However, for two dimension oscillations, the displacement fields are greatly distorted.

The first test of the new procedure used the LIQ element. The representation used in this trial was 100 rectangular elements, each one having the area of two triangular elements. Again, from Table 2 the eigenvalues differ slightly from the analytical values, not as much of the LT elements did, but by the same magnitude of error. Yet, the displacements given by these elements are identical with what is expected from theory. No distortion is noticeable.

The final element tested was the QIQ element. The test basin was divided into 25 elements, each one having the same area as four LIQ or eight LT elements. This procedure showed considerable reduction of the error between the numerical and analytical results, with the first eight eigenvalues having percentage errors of less than two tenths of one percent. The displacement fields had no noticeable distortion.

The test basin results show that for solving for eigenvalues, the QIQ representation is far better with much fewer elements.

The LIQ and QIQ elements do equally well for determining displacement fields. The LT element, for this study, is not a reliable method to use in solving displacements.

The QIQ test case was run on a Cray-1, using about 10,000 words of array storage and taking one second of computation time.

5. APPLICATION OF MODEL TO LAKE ERIE. Lake Erie is the shallowest of the Great Lakes, with an average depth of 60 feet. The lake can be divided into three basins (Figure 7) by reason of geometry and bathymetry. The entire lake is 240 miles long with an average width of 40 miles. The longitudinal axis of the lake is roughly 23 degrees north of east.

Lake Erie, being a basin of varied geometry and bathymetry, does not have analytical solutions for its periods of oscillation. However, spectral analysis of hourly water levels from selected NOAA gages around the lake gives a good indication of how the lake oscillates. For this study, 18 sets of approximately 15-day strings of hourly water levels from Toledo, Marblehead, and Cleveland, Ohio; Erie Pennsylvania; and Buffalo, New York, were analyzed with a resolution of 100 lags and then ensemble averaging by station. Results of this procedure, when the mean lake level was 572.3 feet IGLD (International Great Lakes Datum), are shown in Table 10 along with the results of an earlier study by Platzman and Rao.

Two finite element representations of Lake Erie are used. First the lake is segmented into 92 QIQ elements having a total of 337 nodal points. Next a grid of 368 LIQ elements with 429 nodal points is generated by dividing each QIQ element into fourths. The element representations were done with the major concerns being to adequately resolve the shape of the lake and to increase the density of elements as the bathymetry becomes more varied.

Water depths were digitized at a mean level of 571.0 feet IGLD from the Canadian Hydrographic Service Bathymetric Chart of Lake Erie No. 882 (1971). Changes in the water level from the mean are accounted for by adding or subtracting the difference from the mean at all the nodal points except those along the shore, which are taken to be zero depth at all times.

Results from both element types are shown in Table 3. All the periods computed by the LIQ representation overestimate the averaged lake spectra periods by 3 to 9 percent. Better agreement is reached with the QIQ approach, with errors ranging from -2 to 4 percent difference for the first nine modes of oscillation.

An examination of the results shows that several periods computed spectrally do not appear in the model results, while some modes predicted by the model do not show up at all the stations. The 7.7-hour period at Marblehead and the 4.0-hour period at Cleveland are due to the failure of ensemble averaging to suppress all of the noise found in the individual spectra. Both modes were found to have very low energy. The 5.13-hour modes at Toledo and Erie and the 5.00-hour mode at Cleveland may be due to a combined central and western basin oscillation. The failure of the 9.1-hour mode to appear at Marblehead and Erie can be explained by referring to the displacement plot of Lake Erie's second mode, Figure 11. A nodal line appears at the American shore at Marblehead and near Erie. This is a line of zero oscillation, thus damping out this mode in the spectral results. The 20- to 25-hour mode found at all of the stations is due to tidal effects and is not a free basin oscillation.

Displacement fields from both elements are very similar, with the first five modes all longitudinal oscillations, while the sixth is transverse in nature.

The QIQ representative of Lake Erie required around 40,000 words of array space and took six seconds of execution time on a Cray-1.

6. CONCLUSIONS. A simple, quick, and accurate method for determining the periods of oscillation and relative displacement fields for basins of arbitrary shape and bathymetry has been presented. Comparisons among three types of elements, the LT, LIQ, and QIQ, on a test basin and of two of those elements, the LIQ and QIQ, on Lake Erie show the QIQ element to be superior in representing the domain. Agreement between analytical or statistical and numerical results from a QIQ element representation was shown to be very good. If further work is to be done with this approach, areas of research might include the addition of Coriolis, friction, and nonlinear terms to the equations of motion, along with a suitable boundary condition for forcing oscillations.

7. ACKNOWLEDGEMENTS. This work was done at The Ohio State University as partial requirement of a Master of Science degree for the primary author. I would like to thank the U. S. Army Engineer Waterways Experiment Station and the Office of the Chief of Engineers for permission to publish this manuscript.

8. BIBLIOGRAPHY.

1. Dixon, W. J., Biomedical Computer Programs, Berkeley, Calif.: University of California Press, 1974.
2. Jennings, A., "A direct iteration method of obtaining latent roots and vectors of a symmetric matrix," Proceedings of the Cambridge Philosophical Society, 1967, pp. 755-765.
3. Lamb, Horance, Hydrodynamics. New York: Dover, 1945.
4. Platzman, G. W., and Rao, D. B., "Spectra of Lake Erie water levels," Journal of Geophysical Research, 1964, pp. 2525-2535.
5. Valizadeh-Alavi, H., "Oscillations in Sandusky Bay," Thesis, The Ohio State University, Columbus, Ohio, 1979.

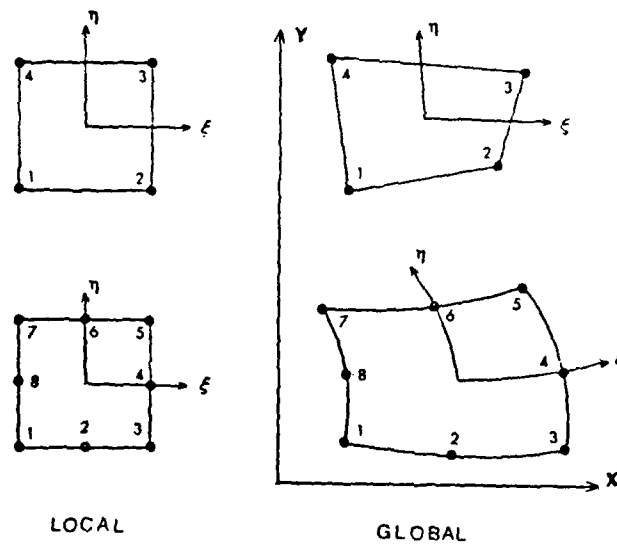


Figure 1. LIQ and QIQ Representations in Local and Global Coordinates

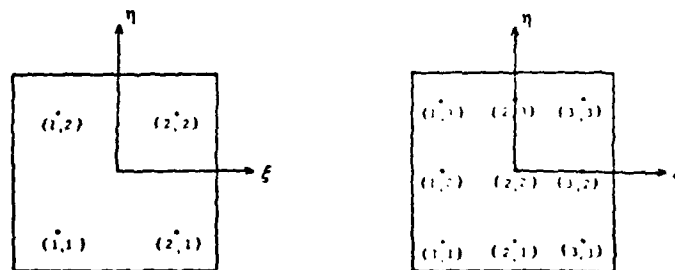


Figure 2. Locations of Gaussian Quadrature Points for Integrating the LIQ and QIQ Elements

Number of Integration Points	Degree of Polynomial	Coordinate Location	Weighting Factor
1	1	$x_1 = 0.0$	$w_1 = 2.0$
2	3	$x_1 = -0.5773503...$ $x_2 = 0.5773503...$	$w_1 = 1.00$ $w_2 = 1.00$
3	5	$x_1 = -0.7745967...$ $x_2 = 0.0$ $x_3 = 0.7745967...$	$w_1 = 0.5555...$ $w_2 = 0.8888...$ $w_3 = 0.5555...$

Table 1. Coefficients of Gaussian Quadrature

Mode Number	(m, n)	Analytical	LT	% Error	LIQ	% Error	QIQ	% Error
1	(0,1)	3.110	3.121	0.35	3.121	0.35	3.111	0.03
2	(1,0)	3.142	3.152	0.31	3.152	0.31	3.142	0.00
3	(1,1)	4.421	4.471	1.13	4.437	0.36	4.421	0.00
4	(0,2)	6.221	6.320	1.59	6.322	1.62	6.231	0.16
5	(2,0)	6.283	6.384	1.61	6.385	1.62	6.293	0.16
6	(1,2)	6.969	7.126	2.25	7.065	1.38	6.979	0.15
7	(2,1)	7.011	7.222	3.01	7.108	1.38	7.020	0.13
8	(2,2)	8.842	9.246	4.57	8.986	1.63	8.859	0.19
9	(0,3)	9.331	9.671	3.64	9.677	3.71	9.401	0.75
10	(3,0)	9.425	9.767	3.64	9.774	3.71	9.495	0.75

Table 2. Comparison of Analytical and Numerical Results for
A 1.00 x Foot Test Basin

Mode	Platzman 569.9	Toledo	Marble	Cleve.	Erie	Buffalo	Average excluding Platzman	LIQ	%E	QIQ	%E
		25.0	25.0	25.0	22.0	25.0					
1	14.35	14.3	14.3	14.3	14.3	14.3	14.3	14.80	3.5	14.25	-0.3
2	9.14	9.1	-	9.1	-	9.5	9.2	9.50	3.3	9.09	-1.2
		-	7.7	-	-	-					
3	5.93	5.88	5.70	5.88	5.41	5.88	5.75	6.25	8.9	6.00	4.3
		5.13	-	5.00	5.13	-					
4	4.15	4.08	4.17	4.25	4.08	4.08	4.13	4.44	7.5	4.25	2.9
		-	-	4.00	-	-					
5		3.64	3.85	3.64	3.70	3.77	3.72	3.86	3.8	3.70	-0.5
6		3.39	3.51	-	-	-	3.45	3.59	4.1	3.41	-1.2
7			3.33	3.33	3.28	-	3.31	3.40	2.7	3.24	-2.1
8		3.23	3.23	3.13	-	3.13	3.18	3.32	4.4	3.17	-0.3
9		2.99	3.03	3.03	3.03	-	3.02	3.18	5.3	3.04	0.7

Table 3. Comparison of Spectra and Model Results at 572.3 Feet IGLD

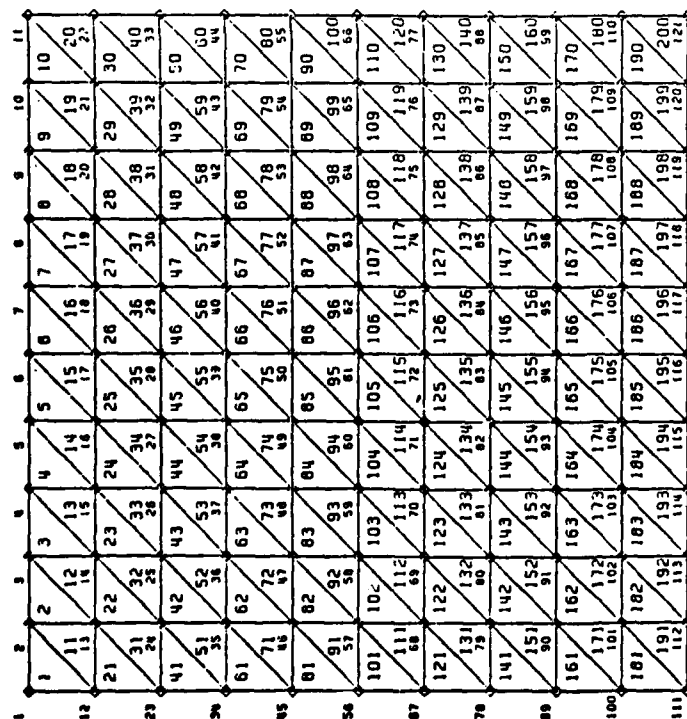


Figure 3. LI Test Basin Finite

Element Representation

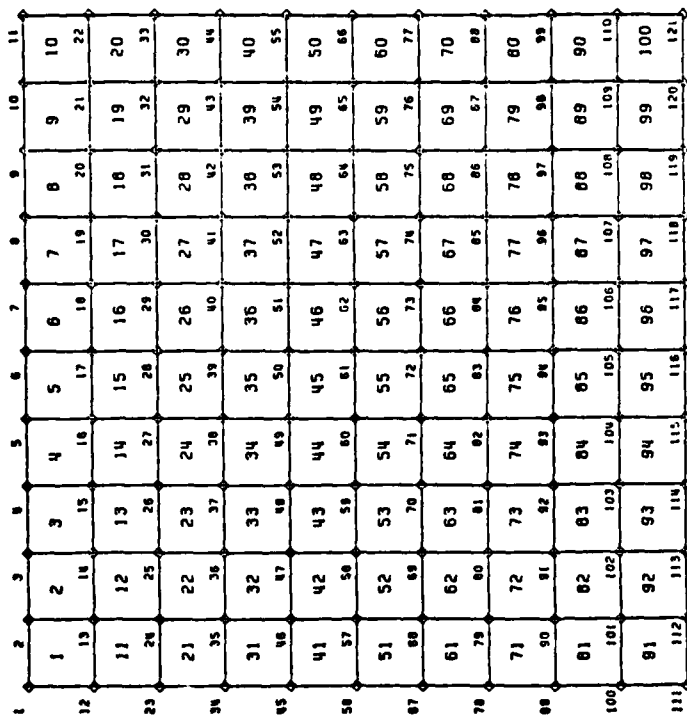


Figure 4. LIQ Test Basin Finite

Element Representation

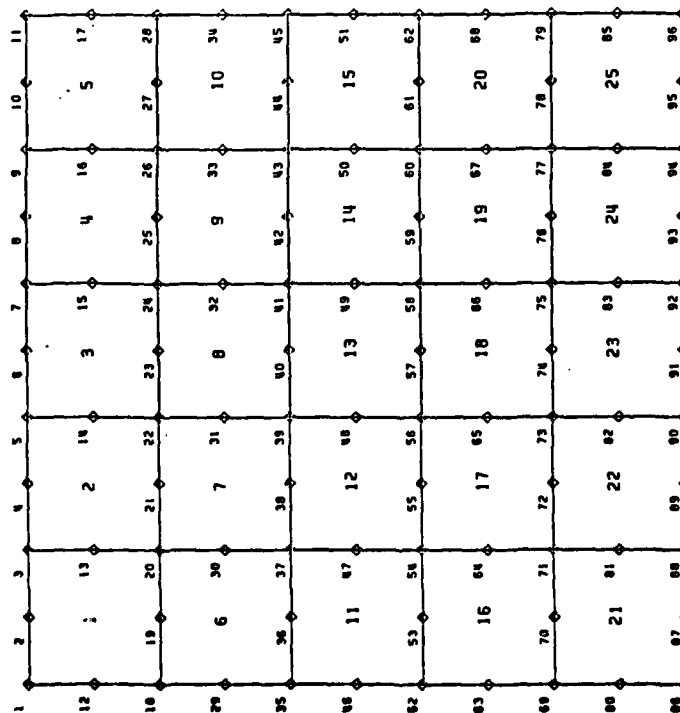


Figure 5. QIQ Test Basin Finite
Element Representation

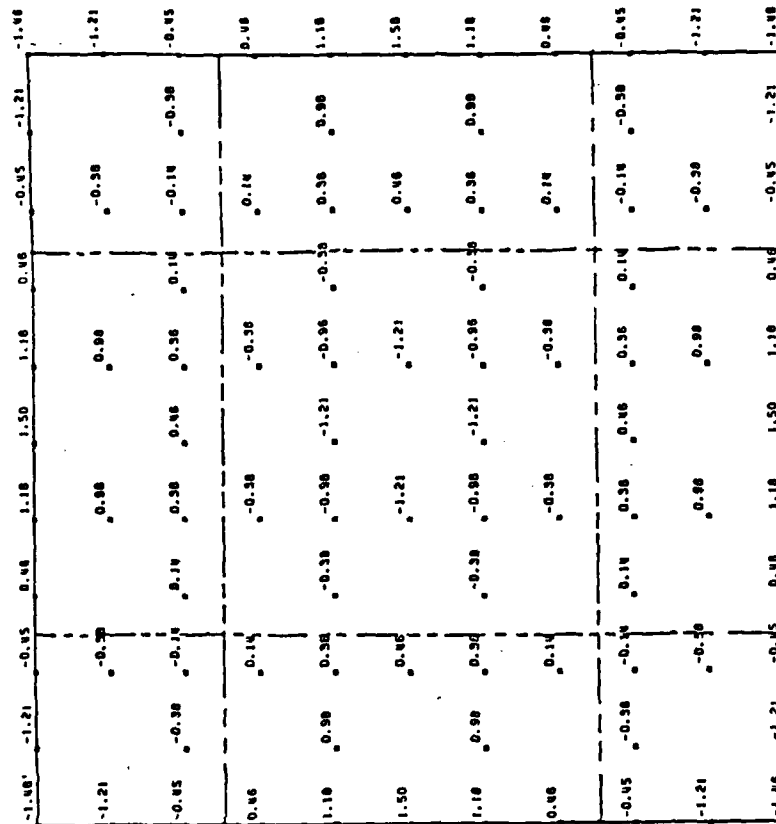


Figure 6. Eighth Mode (2,2) for Test
Basin Using QIQ Elements

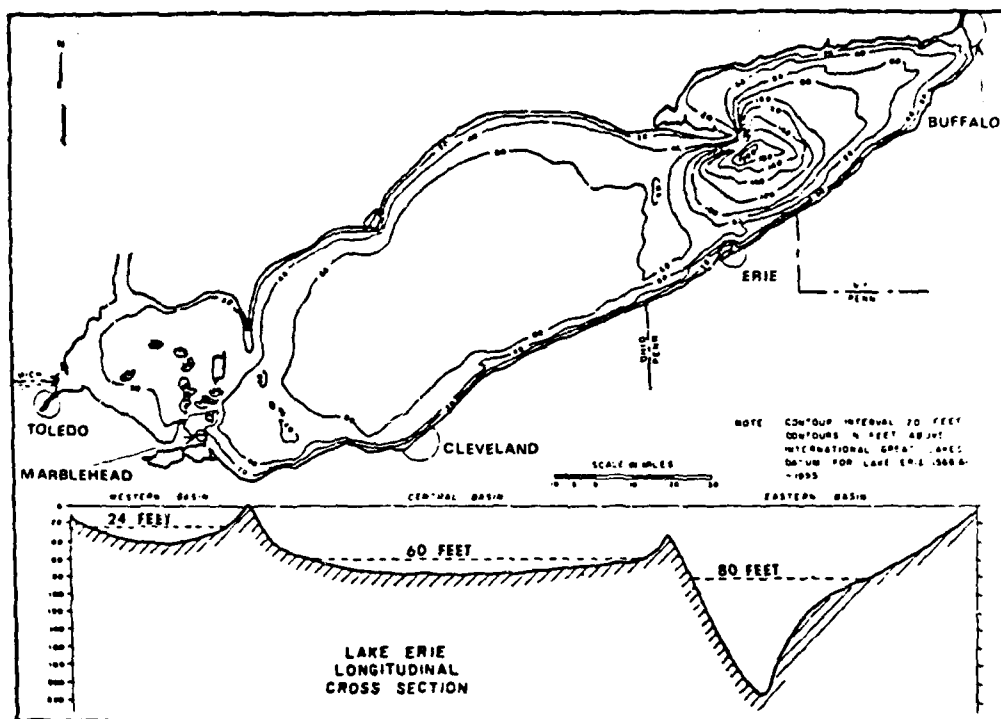


Figure 7. Lake Erie Bottom Topography

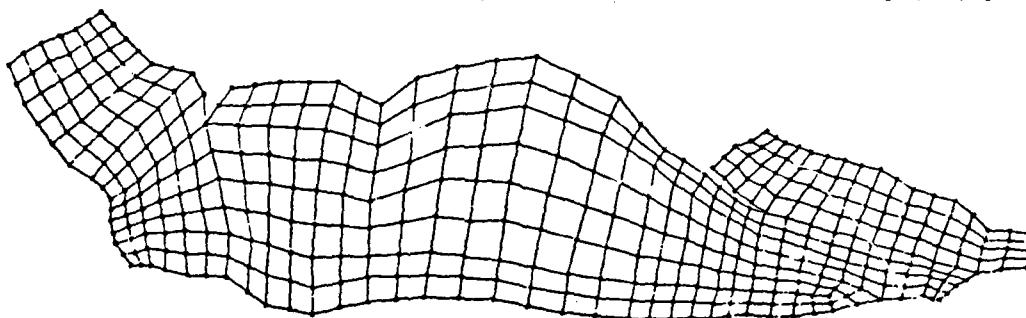


Figure 9. LIQ Element Representation of Lake Erie

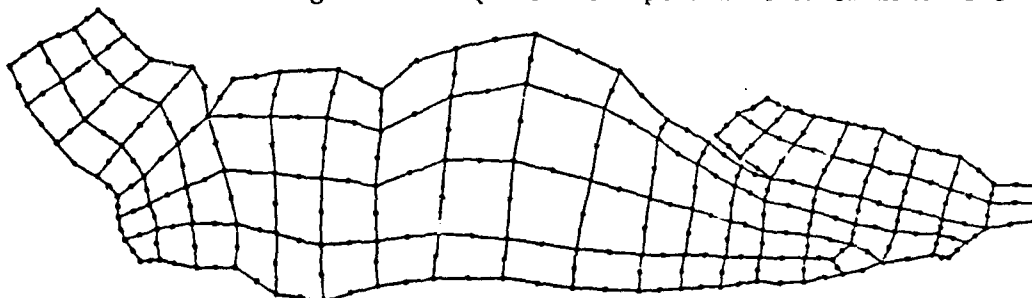


Figure 8. QIQ Element Representation of Lake Erie

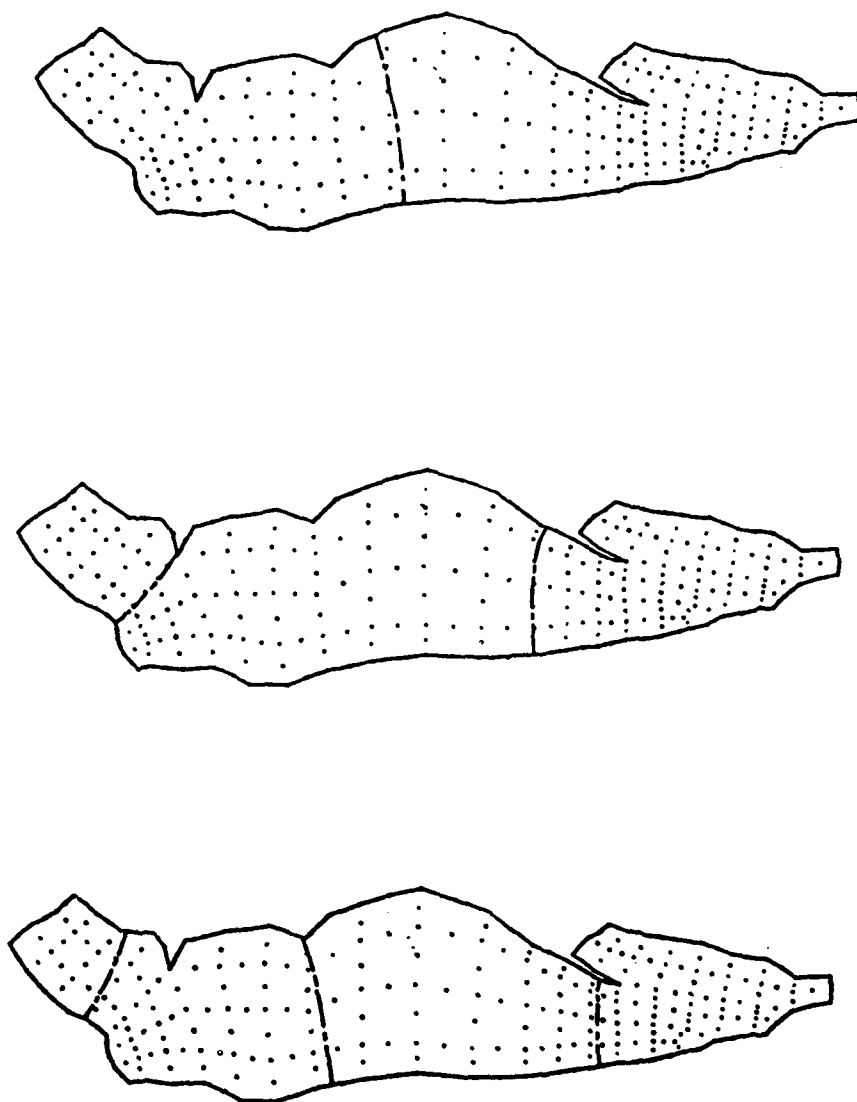


Figure 10. First Three Modes of Lake Erie Using QIQ Elements

A TSUNAMI GENERATION AND PROPAGATION MODEL
DRIVEN BY VERTICAL SEABED MOVEMENTS

Jeff Farickson

U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

Abstract. The generation and propagation of tsunamis by vertical motions of an impermeable seabed is examined in one dimension. Perturbation techniques are used to solve the general equations of fluid motion, and the resulting two nonlinear differential equations describe the effects of seabed motion on the fluid. Near- and far-field wave forms are computed from a finite-difference approximation (FDA) of the differential equations to show possible generation and propagation modes of tsunamis.

The FDA is verified by examining the problems of a solitary wave moving in a uniform depth of fluid, and a solitary wave shoaling upon a plane beach. The computations are shown to agree with analytic solutions and the experiments of others. Results from the numerical model are compared with J. L. Hammack's experiments (1972) with a long wave tank containing a movable bottom section at one end. This comparison shows that the FDA gives uniformly valid results in both the generation region and in the far field.

Introduction. Methods of modeling a tsunami's generation and propagation characteristics have been attempted since the beginning of this century. Due to the repeated destruction of their coastlines by tsunamis, the Japanese proposed some of the earliest generation models. Sano and Hasegawa (1915) studied the simple problem of an instantaneous point disturbance in a three-dimensional fluid of uniform depth, and they used linear wave theory to predict distant wave forms. Syono (1936) considered a finite bottom disturbance in a cylindrical coordinate system. By assuming idealized bottom movement functions, he could predict free-surface wave forms to a limited extent.

The invention of the digital computer, and the 1960 Chilean and 1964 Alaskan earthquakes, induced researchers to develop more sophisticated models. These two earthquakes produced significant tsunamis throughout the Pacific Ocean, giving scientists the first good field data on wave forms and bottom topography changes. L. S. Hwang, David Divoky, and H. L. Butler used two-dimensional finite-difference models of the Pacific Ocean basin to simulate the Alaskan earthquake. In a series of papers (1970, 1972, 1975), they modeled the generation region of the quake with Cartesian coordinates, and the far-field regions with spherical coordinates. The generation model used observed bathymetry changes in the Gulf of Alaska to approximate the disturbance area. Rates of movement were inferred from seismic records. The wave forms created in the generation region model were then used as input to the transoceanic

propagation model. The far-field actions of the wave could then be observed, and the results of the work indicate that the form of a tsunami in the far field is largely independent of the generation mechanism. Their models cannot simulate a tsunami over the generation, transition, and far-field regions without approximate matching methods between different models. Their results also do not address higher order dispersive effects.

J. L. Hammack (1972) studied tsunami generation and propagation mechanisms both theoretically and experimentally. He derived an analytic solution for seabed generated waves by using linear approximations to the incompressible, inviscid equations of motion. His solution is only valid in the wave generation region where nonlinear effects can be considered negligible over certain ranges of bed motion parameters, such as the maximum bottom displacement and the rate of motion. Hammack notes that the wave's nonlinearities grow until the ratio of nonlinear to linear effects approaches unity. This ratio, known as the Ursell (1953) number, defines the far-field or the propagation region. Korteweg and de Vries (1895) derived an equation for this type of wave motion, and Hammack uses solutions to the KdV equation to predict tsunami propagation modes. His analytic solutions indicate that for any initial wave profile whose net volume is finite and positive, a train of solitary waves (first described by Boussinesq 1872) will evolve in the far field. Hammack's analytical method involves asymptotic matching from the linear theory to the KdV equation, and he notes the method's failure to explain the wave's transition region.

Hammack also performed experimental work on wave generation and propagation with a long narrow wave tank. One end of the tank contained a movable block in the bottom, which could be upthrust or downthrown by a servomotor. The motor could be programmed, and Hammack studied two types of block movement. The first was exponential, where the bottom displacement is given by:

$$\zeta(x,t) = \zeta_0 (1 - e^{-\alpha t}) H(b^2 - x^2), \quad t \geq 0 \quad (1)$$

The constant ζ_0 is the maximum (asymptotic) block displacement, H is the Heaviside function, b is the length of the block, and α determines the rate of block motion. For this function, the quantity $(1/\alpha)$, known as the characteristic time t_c , is the time required for the motion to be two-thirds complete. The second type of block motion was transcendental: .

$$\zeta(x,t) = \zeta_0 [1/2(1 - \cos \pi t/T) H(T - t) + H(t - T)] H(b^2 - x^2), \quad t \geq 0 \quad (2)$$

For this case, all of the motion is completed in time T . Hammack

recorded wave profiles generated by these two types of block motion along the length of the wave tank. He studied the characteristics of the waves for different rates of motion, different block lengths, and different total displacements, and he compared his data with his analytic work. The results of his experimental work are used in the present study to verify the numerical calculations and to show how the nonlinear equations presented below give uniformly valid results in both the near and far fields.

Problem Formulation. Recently, researchers such as Hwang and Divoky (1970) have initiated the study of tsunami generation and propagation over uneven sea bottoms with nonlinear theory. However, this work has mostly used Airy theory, and dispersive effects have been ignored. In the related problem of wave generation due to ground movement, nonlinearities have been avoided and work has centered on either constant depth situations (Braddock et al. 1973) or constant slope beaches (Tuck and Hwang 1972). In the following section, the governing equations for long waves generated by ground motion are derived to include effects of dispersion and higher order nonlinearities.

Figure 1 shows an incompressible, inviscid, and irrotational fluid in a constant gravitational field, bounded by a free surface and an impermeable bottom which may change in space and time. Cartesian coordinates are fixed on the quiescent free surface defined by $z = 0$, where z is positive upwards. Laplace's equation governs the fluid motion. If a typical vertical length scale h^* and a typical horizontal length scale L^* are chosen for the fluid domain, a nondimensional length parameter $\epsilon = h^*/L^*$ can be defined. For long-wave problems, ϵ is assumed to be small. By scaling all of the problem variables to the horizontal length scale L^* , the governing equation can be written in nondimensional form as:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0, \quad -h \leq z \leq \eta \quad (3)$$

The free surface kinematic boundary condition can be expressed as:

$$\frac{\partial \eta}{\partial t} + \frac{\partial \phi}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial \phi}{\partial y} \frac{\partial \eta}{\partial y} = \frac{\partial \phi}{\partial z}, \quad z = \eta(x, y, t) \quad (4)$$

The bottom kinematic boundary condition is:

$$\frac{\partial h}{\partial t} + \frac{\partial \phi}{\partial x} \frac{\partial h}{\partial x} + \frac{\partial \phi}{\partial y} \frac{\partial h}{\partial y} + \frac{\partial \phi}{\partial z} = 0, \quad z = -h(x, y, t) \quad (5)$$

The free-surface dynamic boundary condition contains the parameter ϵ in nondimensional form:

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}(\nabla_3 \phi \cdot \nabla_3 \phi) + \frac{\eta}{\epsilon} = 0, \quad z = \eta(x, y, t) \quad (6)$$

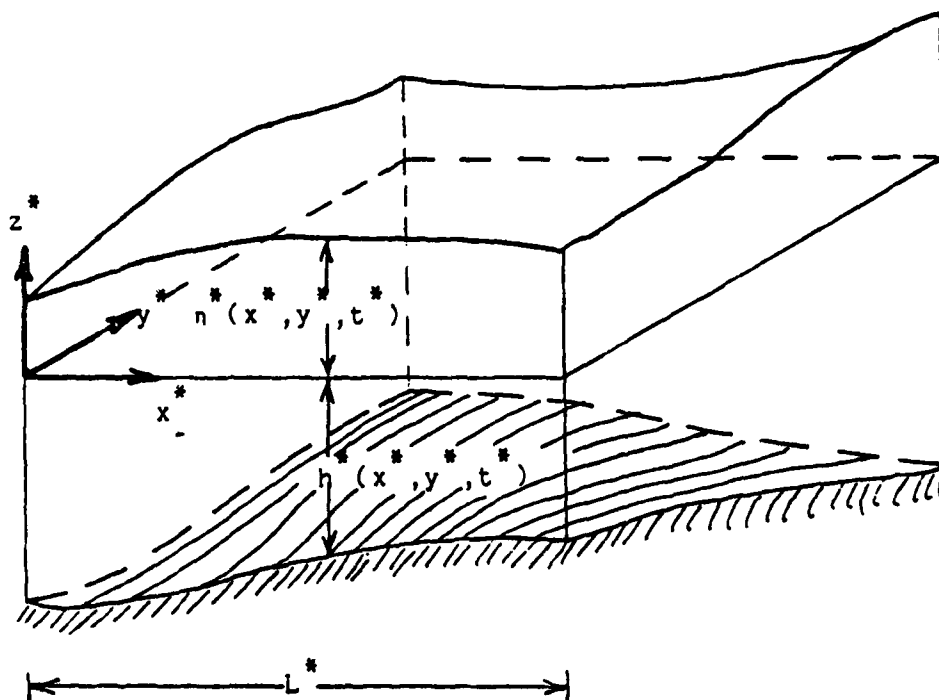


Figure 1
Definition Sketch of the Physical Problem

For long waves, and moderate variations in the bottom profile, the following perturbation solutions can be sought to eliminate the vertical dependency in the velocity potential:

$$\phi(x, y, z, t) = \sum_{n=0}^{\infty} (z + h)^n \phi^{(n)}(x, y, t) \quad (7)$$

This method of perturbation solution was first introduced by Lin and Clark (1959). By substituting the expansion series into Laplace's Equation 3 the following recursive relation in $\phi^{(n)}$ can be found;

$$\phi^{(n+2)} = - \frac{v^2 \phi^{(n)} + 2(n+1) \nabla h \cdot \nabla \phi^{(n+1)} + (n+1) v^2 h \phi^{(n+1)}}{(n+1)(n+2)[(\nabla h)^2 + 1]} \quad (8)$$

Substitution of the series into Equation 5 yields the starting term in the recursive relation:

$$\phi^{(1)} = - \frac{[\partial h / \partial t + \nabla h \cdot \nabla \phi^{(0)}]}{[(\nabla h)^2 + 1]} \quad (9)$$

Following Lin and Clark's argument, the linear terms in Equations 4 and 6 should be comparable in magnitude in order to obtain nontrivial results. Assumptions can be made for the orders of magnitude of the various $\phi^{(n)}$ in terms of the powers of ϵ to satisfy the requirements of long-wave theory and to make the rate of free-surface movement comparable to the rate of bottom motion.

By substituting the perturbation series into Equation 4, and by neglecting terms beyond $O(\epsilon^5)$, one obtains:

$$\frac{\partial \eta}{\partial t} + \nabla \eta \cdot \nabla \phi^{(0)} = \phi^{(1)} + 2(n+h)\phi^{(2)} + 3h^2\phi^{(3)} + 4h^3\phi^{(4)} + O(\epsilon^6) \quad (10)$$

Likewise, by substituting the series into Equation 6 and collecting terms to $O(\epsilon^5)$ yields:

$$\frac{\partial \phi^{(0)}}{\partial t} + \frac{1}{2}[\nabla \phi^{(0)}] + \frac{\eta}{\epsilon} + h \frac{\partial \phi^{(1)}}{\partial t} + h^2 \frac{\partial \phi^{(2)}}{\partial t} = O(\epsilon^5) \quad (11)$$

One can use the recursive relations (Equations 8 and 9) to express $\phi^{(1)}$, $\phi^{(2)}$, $\phi^{(3)}$, and $\phi^{(4)}$ in terms of $\phi^{(0)}$, so that Equations 10 and 11 can be expressed entirely in terms of the free-surface profile η and the depth-averaged horizontal water velocities. Denoting that:

$$\vec{u} = \nabla \phi^{(0)} = (u, v) \quad (12)$$

gives Equations 10 and 11 as:

$$\frac{\partial \eta}{\partial t} + \vec{\nabla}[\vec{u}(\eta + h)] - \frac{h^3}{6} \vec{\nabla} \cdot (\nabla^2 \vec{u}) = \vec{A} \cdot \vec{u} + B \vec{\nabla} \cdot \vec{u} + \frac{3}{2} h^2 \nabla h \cdot \nabla^2 u + C + O(\epsilon^6) \quad (13)$$

and

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{v \eta}{\epsilon} - \frac{h^2}{2} \nabla^2 \left(\frac{\partial \vec{u}}{\partial t} \right) = v \cdot (h \nabla h) \frac{\partial \vec{u}}{\partial t} + 2h \nabla h \cdot \left(\frac{\partial \vec{u}}{\partial t} \right) + D + O(\epsilon^5) \quad (14)$$

where

$$\vec{A} = |\vec{\nabla} h|^2 \nabla h + \frac{1}{2} h^2 \nabla (\nabla^2 h) + 3h \nabla^2 h \nabla h \quad (15)$$

$$B = \frac{3}{2} \vec{\nabla} \cdot (h^2 \nabla h) \quad (16)$$

$$C = \left[|\vec{\nabla} h|^2 + h \nabla^2 h - 2 \nabla h \cdot \vec{\nabla} (\nabla^2 h) - 1 \right] \frac{\partial h}{\partial t} + 2h \nabla h \cdot \nabla (\partial h / \partial t) - \frac{h^2}{2} \nabla^2 (\partial h / \partial t) \quad (17)$$

$$D = \frac{\partial^2 h}{\partial t^2} \nabla h + h \nabla \left(\frac{\partial^2 h}{\partial t^2} \right) \quad (18)$$

The terms C and D are zero for an immovable bottom, and the one-dimensional version of this case yields the equations derived by Mei and LeMehaute (1966). Furthermore, for a horizontal fixed bottom, the right sides of Equations 13 and 14 are zero, yielding the case studied by Lin and Clark (1959).

In a one-dimensional Cartesian system, Equations 13 through 18 reduce to:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x} [u(\eta + h)] - \frac{h^3}{6} \frac{\partial^3 u}{\partial x^3} = Au + B \frac{\partial u}{\partial x} + \frac{3}{2} h^2 \frac{\partial h}{\partial x} \frac{\partial^2 u}{\partial x^2} + C \quad (19)$$

and

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{\epsilon} \frac{\partial \eta}{\partial x} - \frac{h^2}{2} \frac{\partial^3 u}{\partial t \partial x^2} = \frac{\partial}{\partial x} \left(h \frac{\partial h}{\partial x} \right) \frac{\partial u}{\partial t} + 2h \frac{\partial h}{\partial x} \frac{\partial^2 u}{\partial x \partial t} + D \quad (20)$$

where

$$A = \left(\frac{\partial h}{\partial x}\right)^3 + \frac{1}{2} h^2 \frac{\partial^3 h}{\partial x^3} + 3h^2 \frac{\partial^2 h}{\partial x^2} \frac{\partial h}{\partial x} \quad (21)$$

$$B = \frac{3}{2} \frac{\partial}{\partial x} \left(h^2 \frac{\partial h}{\partial x} \right) \quad (22)$$

$$C = \frac{\partial h}{\partial t} \left[\left(\frac{\partial h}{\partial x}\right)^2 + h \frac{\partial^2 h}{\partial x^2} - 2 \frac{\partial h}{\partial x} \frac{\partial^3 h}{\partial x^3} - 1 \right] + 2h \frac{\partial h}{\partial x} \frac{\partial^2 h}{\partial x \partial t} + \frac{h^2}{2} \frac{\partial^3 h}{\partial x^2 \partial t} \quad (23)$$

$$D = \frac{\partial^2 h}{\partial t^2} \frac{\partial h}{\partial x} + h \frac{\partial^3 h}{\partial x \partial t^2} \quad (24)$$

The Finite-Difference Approximation (FDA). The approximation to Equations 19 and 20, as well as the computational method used here, follows the scheme developed by D. H. Peregrine (1967) for a somewhat simpler set of equations. Peregrine addresses the problem of long waves shoaling on a plane beach where no bottom motion occurs. His solution scheme uses forward time differences for the continuity equation and backward time differences for the momentum equation. Both central and Crank-Nicholson spatial differences are employed. His computational method calculates a provisional free-surface profile at the advanced time level with the continuity approximation (which is explicit in η). The provisional η is then used to calculate the horizontal water velocities at the advanced time level. The water velocities are then averaged over the present and advanced time levels, and this information is used to calculate the final wave profile at the advanced time level. Then the time-step advances, and the process repeats. Peregrine had shown the stability and accuracy of this method in an earlier paper (1966) by simulating the movement of a solitary wave along a constant depth channel and noting the wave's degradation due to numerical error.

When Peregrine's method of finite-differencing is applied to the continuity Equation 19, one obtains:

$$\begin{aligned}
\eta(i, j+1) = \eta(i, j) + \Delta t \left(\frac{1}{12} \left(\frac{h}{\Delta x} \right)^3 u(i+2, j) + \left\{ \frac{B}{2\Delta x} \right. \right. \\
\left. \left. + \frac{3}{2} \left(\frac{h}{\Delta x} \right)^2 \left(\frac{\partial h}{\partial x} \right)_{i, j} - \frac{1}{6} \left(\frac{h}{\Delta x} \right)^3 \right. \right. \\
\left. - \frac{1}{2} [\eta(i, j) + h] \frac{1}{\Delta x} \right\} u(i+1, j) + \left\{ A - 3 \left(\frac{h}{\Delta x} \right)^3 \left(\frac{\partial h}{\partial x} \right)_{i, j} \right. \\
\left. - \left(\frac{\partial h}{\partial x} \right) - [\eta(i+1, j) - \eta(i-1, j)] \frac{1}{2\Delta x} \right\} u(i, j) \\
+ \left\{ \frac{1}{2\Delta x} [\eta(i, j) + h] + \frac{1}{6} \left(\frac{h}{\Delta x} \right)^3 - \frac{B}{2\Delta x} \right. \\
\left. + \frac{3}{2} \left(\frac{h}{\Delta x} \right)^2 \left(\frac{\partial h}{\partial x} \right)_{i, j} \right\} u(i-1, j) \\
\left. + \left[\frac{-1}{12} \left(\frac{h}{\Delta x} \right)^3 u(i-2, j) \right] + C \right) \quad (25)
\end{aligned}$$

The momentum equation (20) becomes:

$$\begin{aligned}
\left[\frac{u(i, j)}{4\Delta x} + M(i, j) \right] u(i+1, j+1) + N(i, j) u(i, j+1) \\
+ \left[P(i, j) \frac{u(i, j)}{4\Delta x} \right] u(i-1, j+1) \\
= \left[M(i, j) - \frac{u(i, j)}{4\Delta x} \right] u(i+1, j) + N(i, j) u(i, j) + \left[P(i, j) \right. \\
\left. + \frac{u(i, j)}{4\Delta x} \right] u(i-1, j) + D \\
+ \frac{[\eta(i-1, j+1) - \eta(i+1, j+1) + \eta(i-1, j) - \eta(i+1, j)]}{4\epsilon\Delta x} \quad (26)
\end{aligned}$$

where

$$M(i, j) = - \left[\frac{1}{2} \left(\frac{h}{\Delta x} \right)^2 + \left(\frac{h}{\Delta x} \right) \left(\frac{\partial h}{\partial x} \right)_{i, j} \right] \frac{1}{\Delta t} \quad (27)$$

$$N(i,j) = \left[1 + \left(\frac{h}{\Delta x} \right)^2 - \left(\frac{\partial h}{\partial x} \right)_{i,j}^2 - h \left(\frac{\partial^2 h}{\partial x^2} \right)_{i,j} \right] \frac{1}{\Delta t} \quad (28)$$

$$P(i,j) = \left[\left(\frac{h}{\Delta x} \right) \left(\frac{\partial h}{\partial x} \right)_{i,j} - \frac{1}{2} \left(\frac{h}{\Delta x} \right)^2 \right] \frac{1}{\Delta t} \quad (29)$$

Here, i denotes the location of the space gridpoint, while j denotes the time level. The variables A , B , C , D , and h are always evaluated at (i,j) , and from a programming standpoint they can be viewed as "constants." In FORTRAN code, these five variables can be represented as FUNCTION subprograms to simplify program writing. Equation 25 is explicit in η , while the form of Equation 26 yields a tridiagonal matrix for the advanced time level of u .

Results. After Equations 25 through 29 were coded into FORTRAN, the stability and accuracy of the FDA was tested with two simple bottom profiles. The movement of a solitary wave over a region of uniform depth was studied, since an analytic solution to this case is known. The case of a solitary wave shoaling upon a plane beach was also examined and computations were compared with the work of others. No bottom motion was considered in either test.

Laitone (1963) has shown that the solitary wave solution of Boussinesq (1872) is an approximate solution to higher order long-wave theories, and it is known that a solitary wave is a limiting case to cnoidal wave solutions. This wave, also called a soliton, retains a permanent shape in a Lagrangian coordinate system. In nondimensional form the free-surface profile of a soliton can be expressed as:

$$\eta(x,t) = H \operatorname{sech}^2 \left[\sqrt{\frac{3H}{4\epsilon}} \frac{(x - Ct)}{\epsilon} \right] \quad (30)$$

where H is the wave amplitude and C is the wave speed, given as:

$$C = \left(1 + \frac{H}{\epsilon} \right)^{1/2} \quad (31)$$

The depth-averaged horizontal water velocity is

$$u(x,t) = \frac{C\eta(x,t)}{\epsilon + \eta(x,t)} \quad (32)$$

These equations were put into the computer code as initial conditions, and the amount of wave degradation over time was noted for changes in the FDA's spatial resolution and time-step.

Two points must be noted about the use of a soliton in the

calculation scheme. Due to the infinite wavelength of a soliton and the uniform depth in this problem, no horizontal length scale L^* exists. The use of the parameter ϵ in Equations 30 through 32 must be explained for the solitary wave problem. Following Peregrine's scaling methods (1966), the dimensional form of Equations 19 and 20 can be scaled in terms of the constant depth. For the case of a uniform depth, it can be shown that this scaling gives Equations 19 and 20 when ϵ equals one. The infinite wavelength of Equation 30 must also be made to fit into a computational grid of finite length. Madsen and Mei's work (1969) suggests setting the free-surface elevation to zero when it falls below some small fraction of the wave amplitude H . The approach, in effect, chops off the leading and trailing edges of the wave. The mass and momentum lost by this approximation degrades the solution, however, and the discontinuities at the "ends" of the wave emit noise spikes of $2\Delta x$ width. The ratio of the cutoff height to the wave amplitude plays as important a part in the stability tests as the spatial resolution and time-step. Madsen and Mei use a cutoff ratio of 0.001 for their computations, a ratio which gives satisfactory results for the present computational scheme, but smaller ratios give even better results.

Figure 2 illustrates the accuracy of the computational method for a small cutoff ratio ϵ and moderate values of Δx and Δt . This calculation advances 1000 time-steps with a minimum of numerical degradation. A series of computations with a constant cutoff ratio confirmed Peregrine's (1966) claim that the computational method remains stable and accurate for $\Delta t/\Delta x$ ratios up to one.

Figures 3 and 4 illustrate the FDA's ability to model wave shoaling upon a plane beach. Figure 3 shows a time series of free-surface profiles for a shoaling wave. Both Peregrine (1967) and Madsen and Mei (1969) perform calculations for a soliton shoaling on a plane beach, and their results are compared with the present computations in Figure 4. While the changes in wave amplitude over depth agree for the three computations, some differences exist in initial and boundary conditions. Madsen and Mei use the bottom profile illustrated in Figure 3, but their initial condition places the leading edge of the soliton over the toe of the beach. Peregrine does not use a uniform depth region to begin his computation but places his initial wave over the beach slope. He then scales his problem according to the depth beneath the initial wave, so that $\epsilon = 1$ at zero time. In the present computations, the wave starts entirely in a uniform depth region. These differences in initial conditions explain the variations in shoaling rate at the toe of the beach in Figure 4.

For the simulation of Hammack's wave tank experiments with the FDA, the bottom motion functions (Equations 1 and 2) are nondimensionalized with the block length b as the horizontal length scale. In this form, the block length becomes one, and the parameter ϵ expresses the ratio of the water depth to the block length. Hammack studied the waves

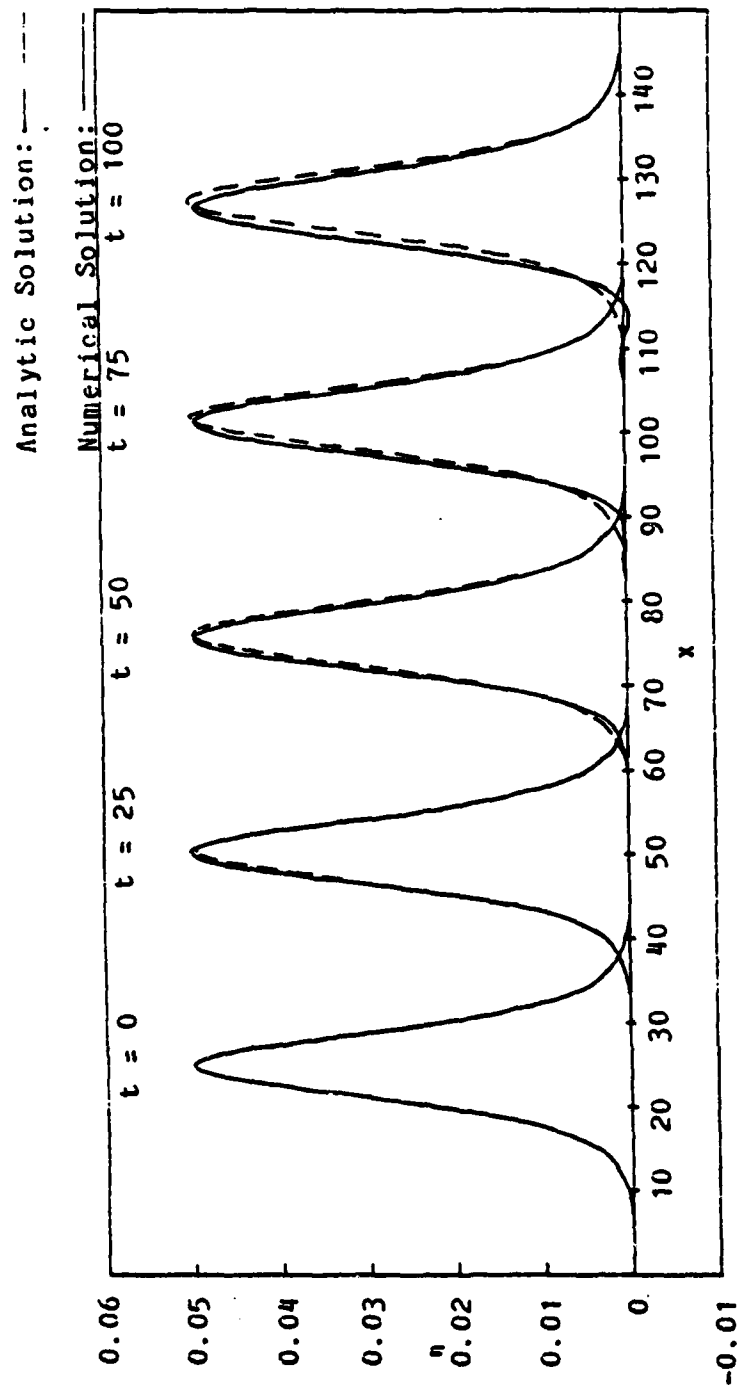


Figure 2
 Solitary Wave Free-Surface Profiles
 Analytical Solution vs. Numerical Calculations:
 $\Delta t = 0.1$ $\epsilon = 1.0$
 $\Delta x = 0.5$ $c = 10.0$
 $\Delta t / \Delta x = 0.2$ tank length = 250.0

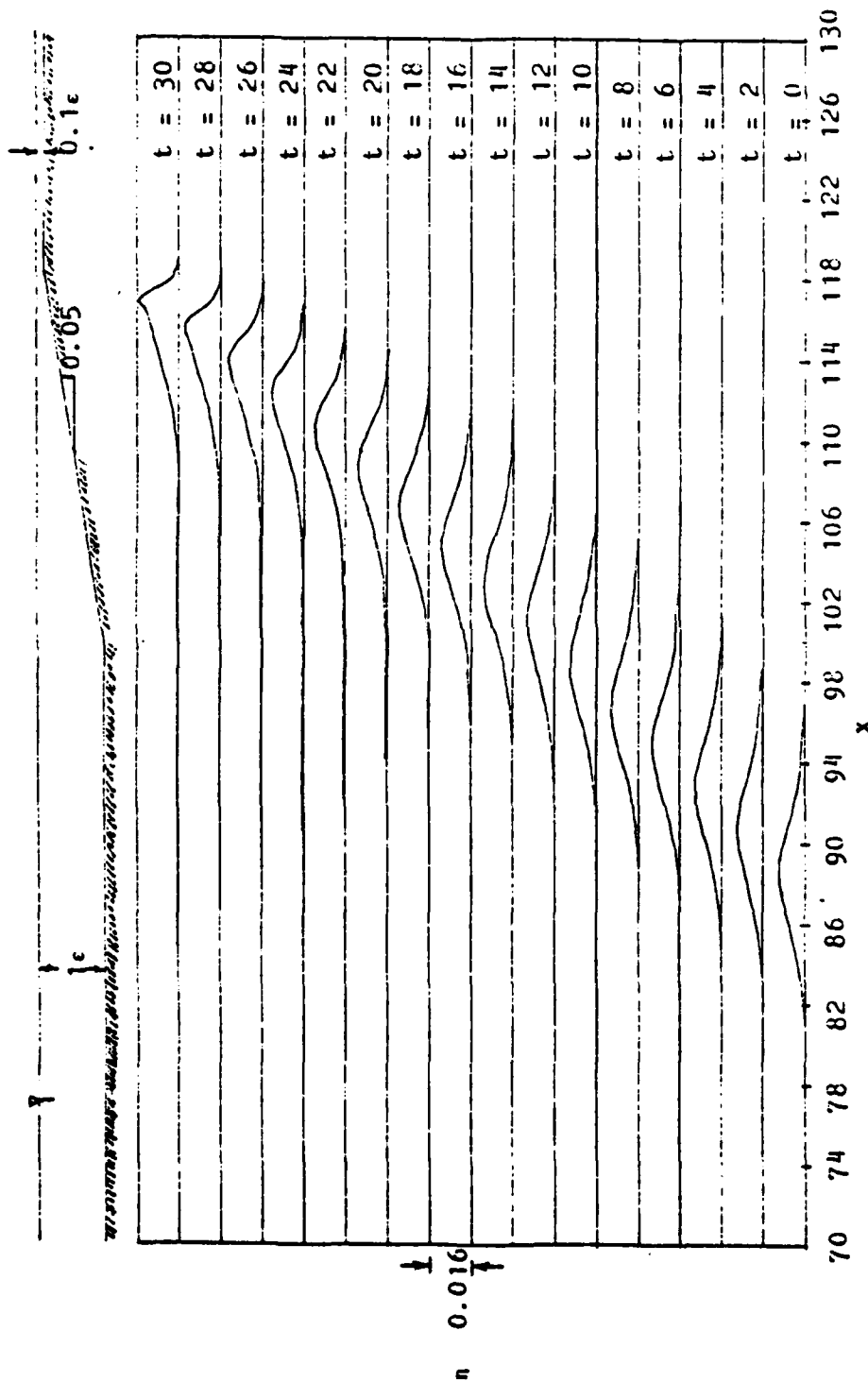


Figure 3
Solitary Wave Shoaling upon a Beach
of Uniform Slope = 0.05
 $\Delta t = 0.125$ $\Delta x = 0.25$
 $\Delta t / \Delta x = 0.5$ $c = 10^{-5}$

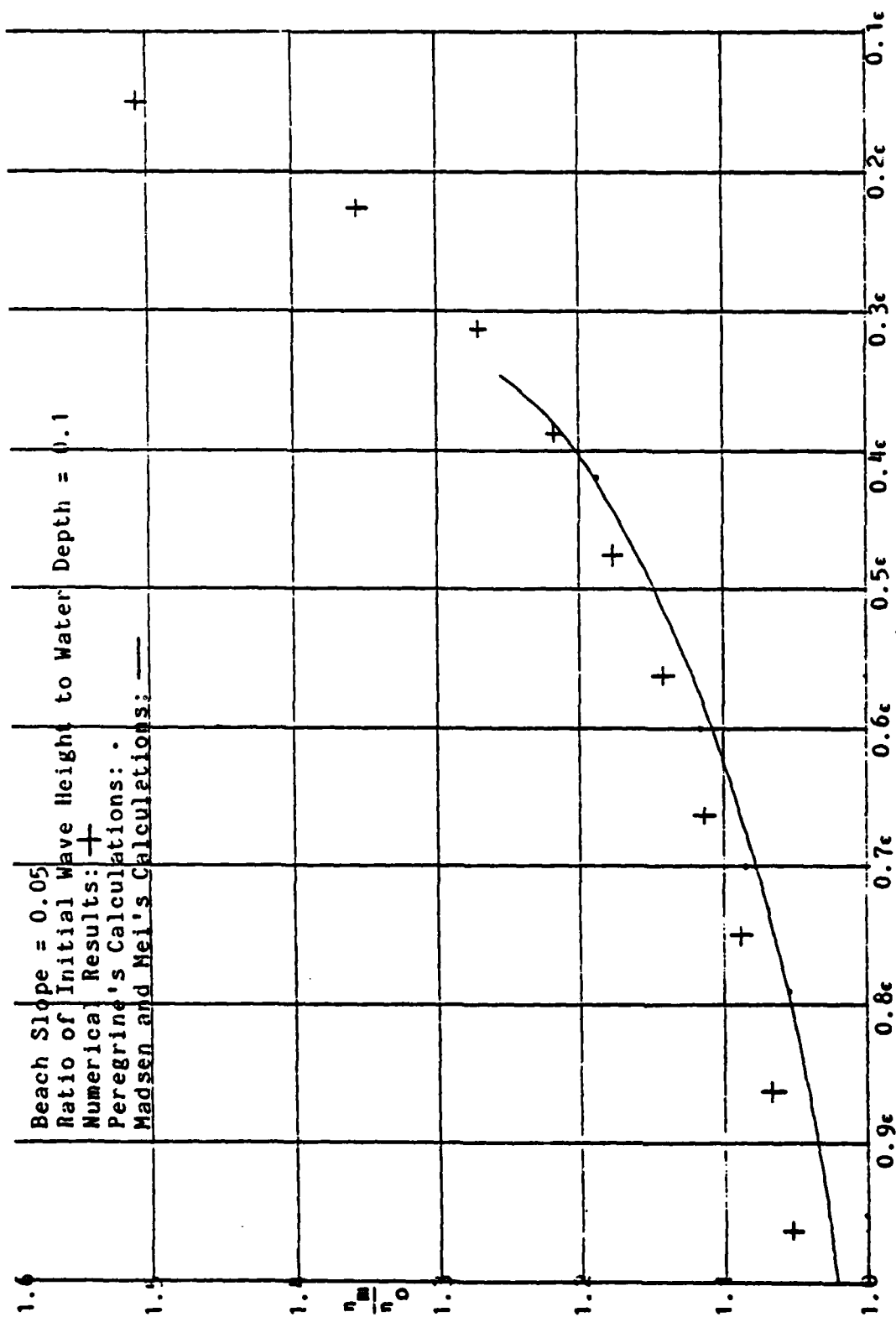


Figure 4
 Variation of Wave Amplitude vs. Water Depth for a Shoaling Wave

resulting from three rates of block motion which he termed "impulsive," "transitional," and "creeping" motion. Impulsive motion occurs when the block rises rapidly enough to cause the fluid above it to rise as a solid body during the block motion. Creeping motion allows the displaced fluid to flow off of the block during the time of motion, while transitional motion lies between these extremes. The use of the FDA in simulating Hammack's results is constrained by the condition that $\epsilon \ll 1$. Hammack performed all three rates of motion for Equation 1 with a small ϵ , while only the impulsive rate experiment for Equation 2 used a small ϵ . These four cases are simulated with the FDA, and the calculations are compared with the experimental results in the generation region. For these four calculations, a Δx of 0.02 is used, which models the block in 50 space steps. This meshwidth proves to give a fine numerical resolution at a reasonable computer cost.

The simulation of Hammack's experiments requires the use of the bottom motion functions (Equations 1 and 2) in the computer code, as well as higher derivatives in space and time. The depth functions and its derivatives appear in Equations 19 through 24 and appear in the computer code as FUNCTION subprograms. Due to the Heaviside functions in Equations 1 and 2 (or due to the leading vertical edge of the block, from a physical point of view) many of the bottom function derivatives cannot be evaluated at $x = 1$. This difficulty required the placement of an internal boundary condition into the code to evaluate η and u at this point. Initial simplistic attempts to evaluate this boundary condition by setting the higher derivatives to zero caused a numerical "smearing" at the discontinuity, which generated numerical noise. The internal boundary condition problem was finally resolved by considering mass and momentum conservation in a Δx region about the edge point. Values for η and u at $x = 1$ were derived from information at the nearest meshpoints by using conservation principles. When this internal boundary condition was used, the numerical noise ceased.

Figures 5, 6, and 7 illustrate the agreement between the computations and Hammack's results in the generation region for the three rates of experimental block upthrust. The figures show the change in the free surface over time at the block edge abutting the end of the wave tank ($x = 0$) and at the leading edge ($x = 1$). The largest differences in results appear in the impulsive motion, which can be considered the worst case numerically. Figures 8, 9, and 10 present time series of the wave shapes from the three rates of bottom motion. In the impulsive case, the body of water displaced by the block begins to break into a propagating wave train fairly close to the block, while the wave from the creeping motion appears as an undular bore in the generation region. Peregrine (1966) shows how an undular bore forms into a solitary wave train as it propagates.

Figures 11 and 12 show motion histories and a time series of wave profiles for impulsive motion with Equation 2. As with the case of Figure 5, Figure 11 illustrates that the water above the block moves as

Hammack's Experiments: — — — —

Numerical Results: —————

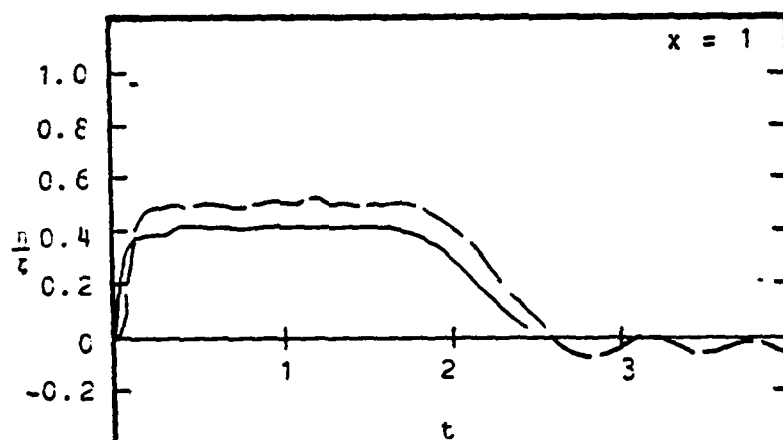
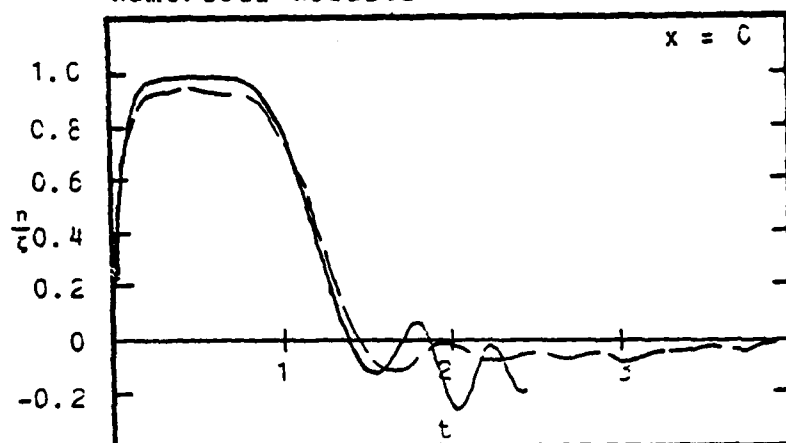


Figure 5
Free-Surface Fluctuations at $x=0$ and $x=1$ vs. Time:
Numerical Calculations vs. Hammack's Data, for an
Impulsive Exponential Block Upthrust

$\Delta t = 0.002$ $t_0 = 0.0164$
 $\Delta x = 0.02$ $a_0 = 16.0869$
 $\epsilon = 0.082$ $t_c = 0.069$

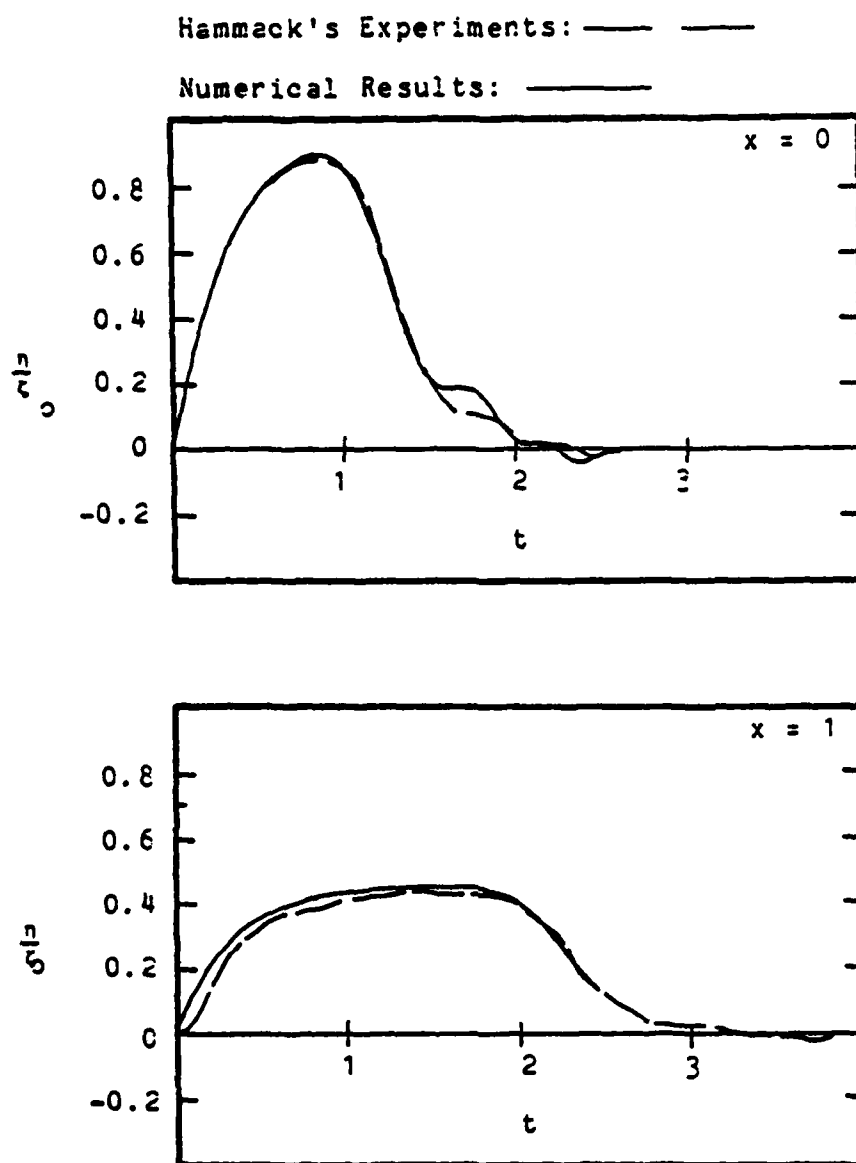


Figure 6
Free-Surface Fluctuations at $x = 0$ and $x = 1$ vs. Time:
Numerical Calculations vs. Hammack's Data, for a
Transitional Exponential Block Upthrust

$\Delta t = 0.002$ $\tau_0 = 0.0082$
 $\Delta x = 0.02$ $a_0 = 2.8461$
 $c = 0.0820$ $t_0 = 0.39$

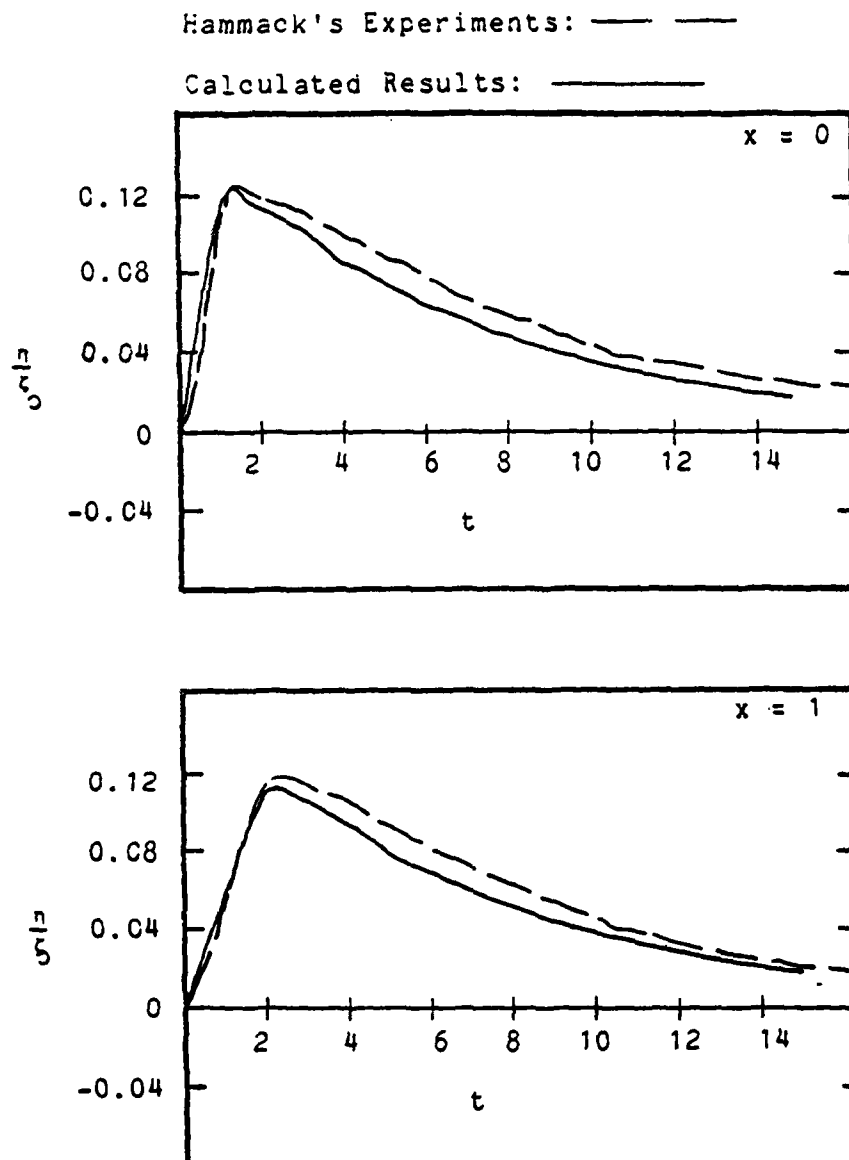


Figure 7
 Free-Surface Fluctuations at $x=0$ and $x=1$ vs. Time:
 Numerical Calculations vs. Hammack's Data, for a
 Creeping Exponential Block Upthrust

$\Delta t = 0.01$	$\tau_0 = 0.0246$
$\Delta x = 0.1$	$\alpha = 0.1275$
$\epsilon = 0.0820$	$t_c = 8.70$

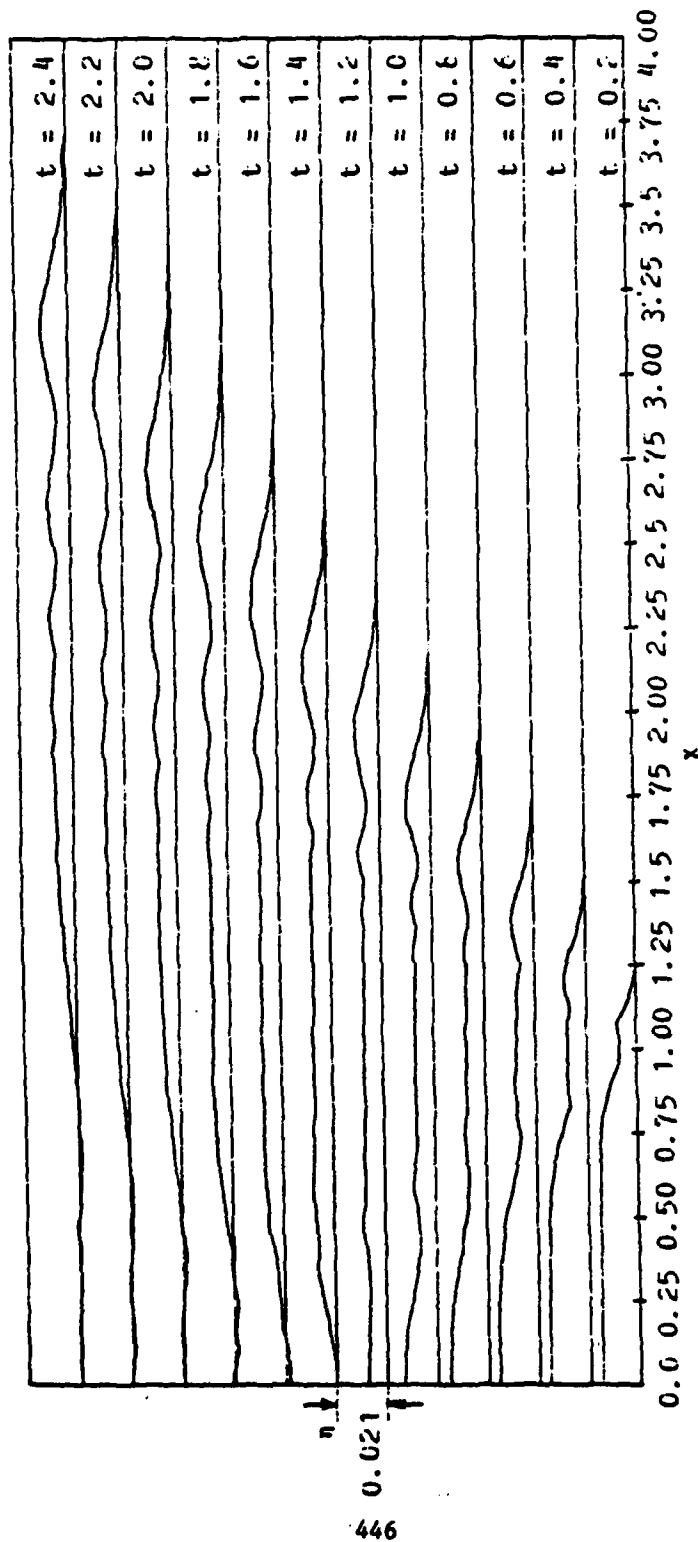


Figure 8
Calculated Free-Surface Profiles over Time:
Impulsive Exponential Block Upthrust

$\Delta t = 0.002$ $\zeta_0 = 0.0164$
 $\Delta x = 0.02$ $\alpha_0 = 16.0869$
 $\epsilon = 0.082$ $t_c = 0.069$

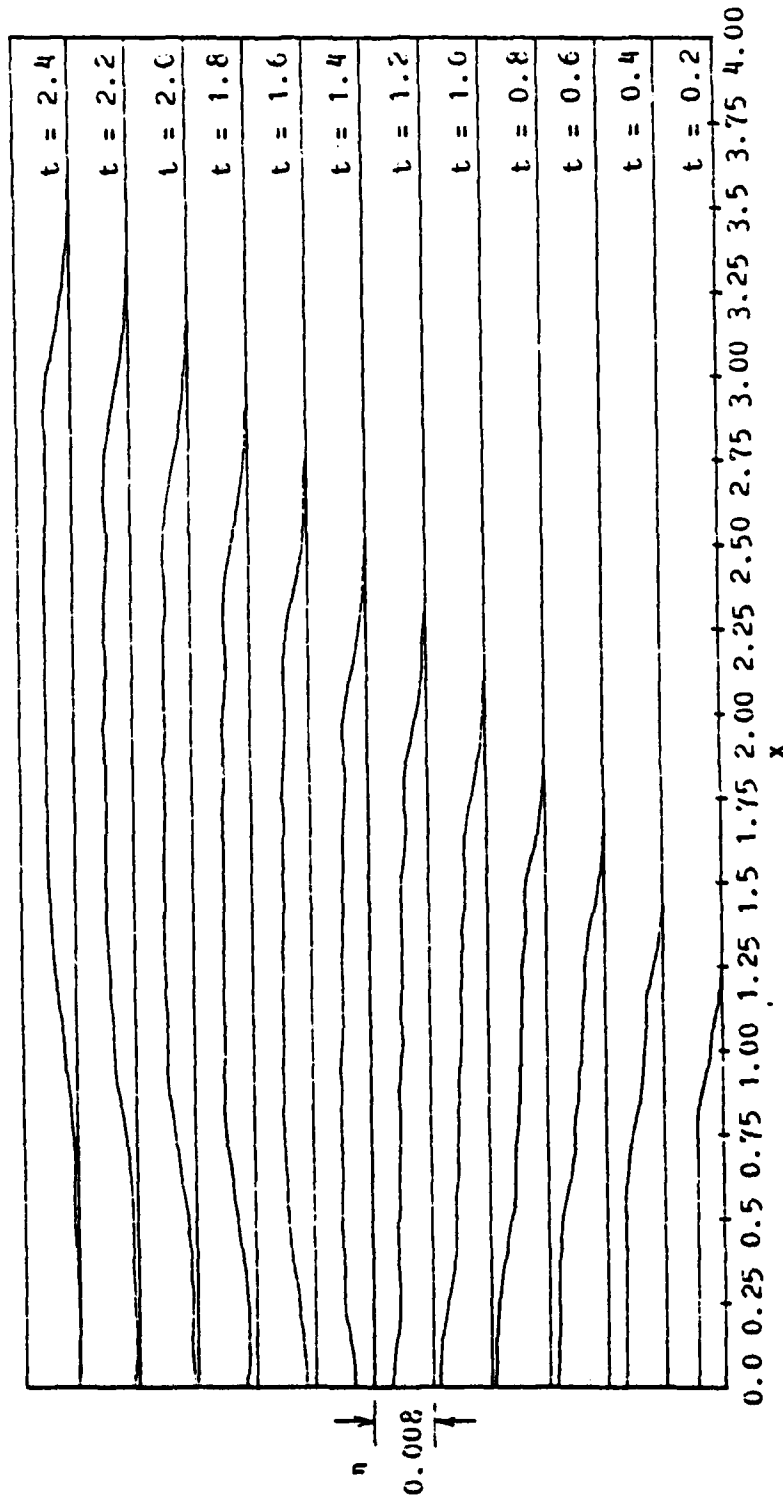


Figure 9
Calculated Free-Surface Profiles over Time:
Transitional Exponential Block Upthrust

$$\begin{aligned} \Delta t &= 0.002 & \zeta_0 &= 0.0082 \\ \Delta x &= 0.02 & \alpha &= 2.8461 \\ \epsilon &= 0.0820 & l_c &= 0.39 \end{aligned}$$

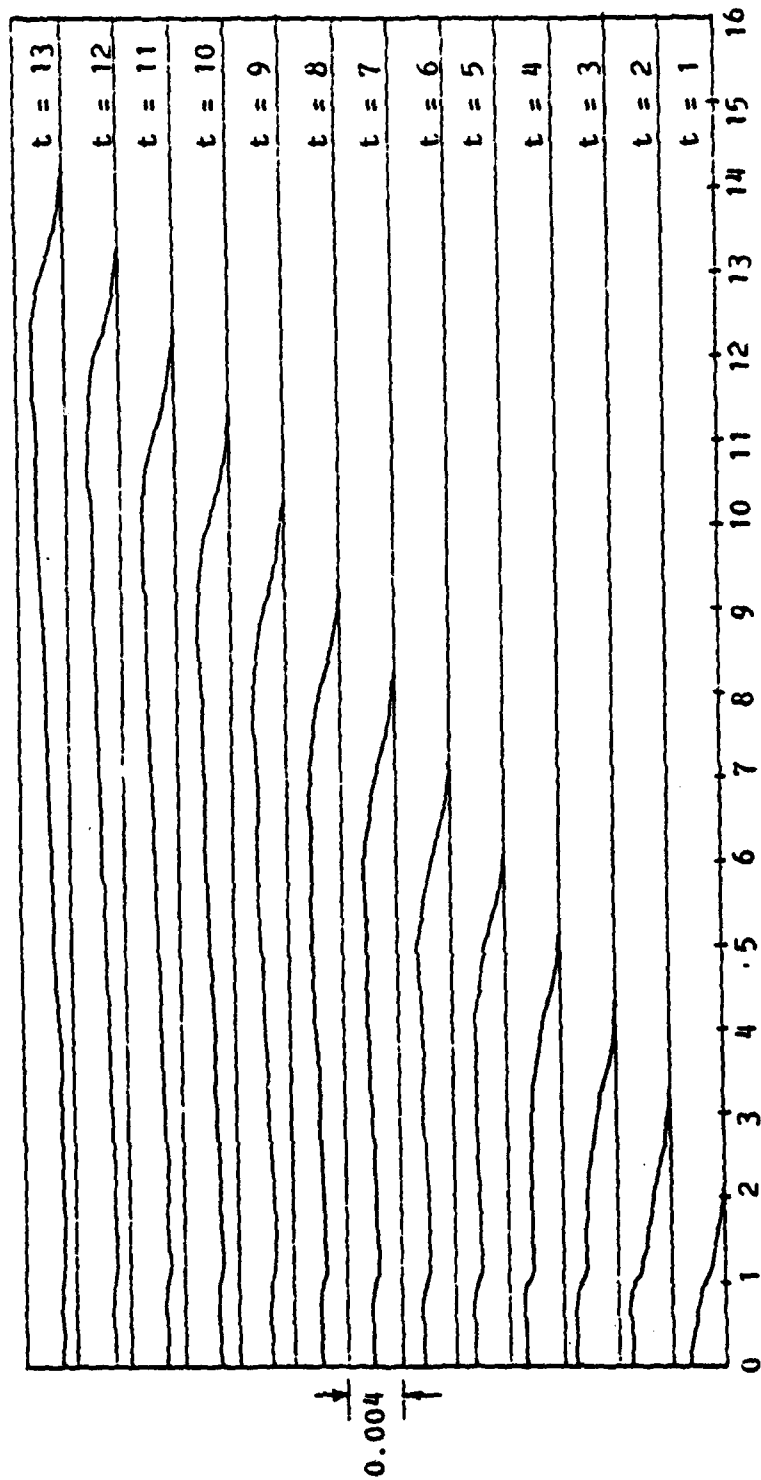


Figure 10
Calculated Free-Surface Profiles over Time:
Creeping Exponential Block Upthrust

$$\begin{aligned} \mu &= 0.01 & \zeta_0 &= 0.0246 \\ \Delta x &= 0.1 & \alpha &= 0.1275 \\ \epsilon &= 0.0820 & t_c &= 8.70 \end{aligned}$$

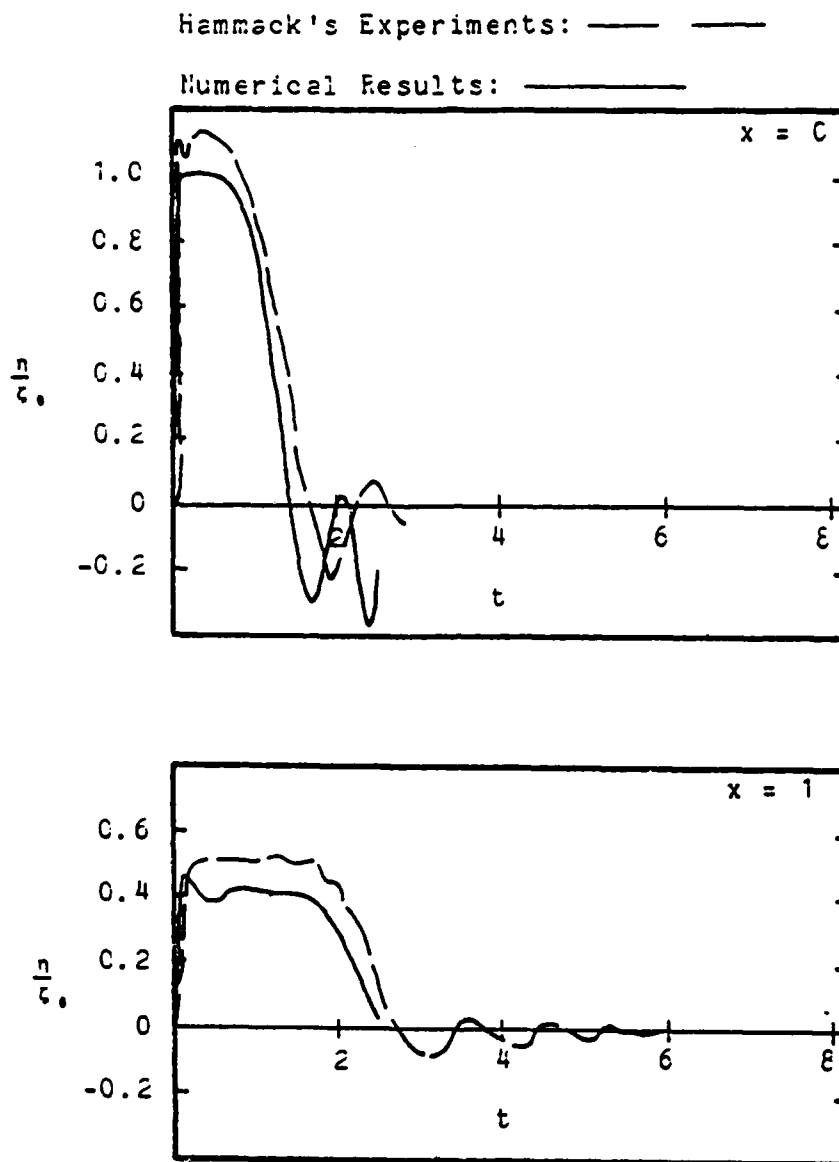


Figure 11
Free-Surface Fluctuations at $x=0$ and $x=1$ vs. Time:
Numerical Calculations vs. Hammack's Data, for an
Impulsive Half-Sine Block Upthrust

$$\begin{aligned} \Delta t &= 0.002 & \zeta_0 &= 0.0164 \\ \Delta x &= 0.02 & \alpha &= 0.13 \\ \epsilon &= 0.1639 & t_c &= 0.13 \end{aligned}$$

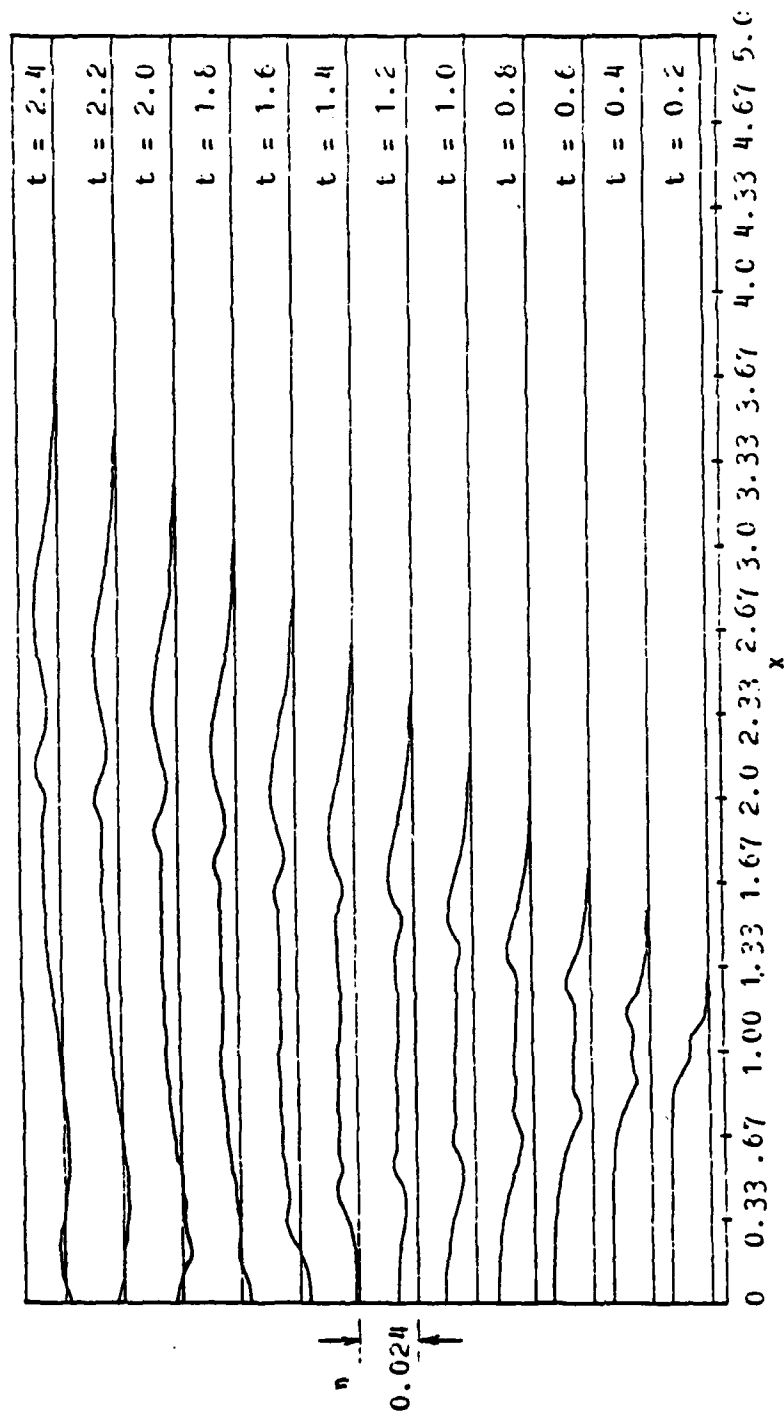


Figure 12
Calculated Free-Surface Profiles over Time:
Impulsive Half-Sine Block Upthrust

$$\begin{aligned} \Delta t &= 0.002 & \zeta_0 &= 0.0164 \\ \Delta x &= 0.02 & \alpha_0 &= 0.13 \\ \epsilon &= 0.1639 & t_c &= 0.13 \end{aligned}$$

a solid body at the beginning of motion. The calculated curve at $x = 0$ attains a height of $\eta/\zeta_0 = 1$, where ζ_0 is the maximum block displacement. Hammack's results indicate an even higher elevation at the center of the block, which is not physically possible from momentum considerations. Hammack notes that his high values are caused by air entrainment into the water from a faulty seal at the edge of the block. Figure 12 illustrates how the displaced water mass begins to form an undulating wave train as it moves away from the block.

The FDA's ability to model both the wave generation near the block, and its propagation into the far field where nonlinear behavior predominates, is illustrated in Figures 13 through 15. While the generation region calculations of Figures 5 through 12 use a meshwidth of $\Delta x = 0.02$, computer costs require a Δx of 0.1 for the calculations in Figures 13 through 15. This increase in step size is due to the large propagation distances and wave travel times needed to model far-field propagation. This coarse step size causes the periodic oscillations at $x = 0$ and $x = 1$ in the numerical results. As the wave moves into the propagation region, however, its shape agrees with Hammack's measurements. The calculations at $x = 2.64$ match closely, and a solitary wave train evolves in the far field (Figure 14).

A comparison of computational and experimental results in the far field shows an increasing divergence in wave speeds and amplitudes. By $x = 33.8$, the leading computed soliton stands approximately 40 percent taller than its physical counterpart, and in accordance with the Boussinesq celerity Equation 31 the numerical waves outrun the experimental results. These differences in wave height can be attributed to viscous energy dissipation along the tank walls and at the free surface. Hammack cites the study by Keulegan (1948) in his analysis of wave decay along the length of the tank. Keulegan derives a formula for determining the downstream height of a solitary wave in terms of its initial height, the water depth, the tank width, and the kinematic viscosity of the fluid. Hammack applies this formula to his experimental data to show that the decay rates of his waves fall within the 40 percent amplitude decay range predicted by Keulegan.

Figure 15 shows a time series of wave profiles for this calculation. The formation of a solitary wave train from the initial displaced water mass is illustrated, and this figure shows how the wave train breaks into isolated solitons in the far field.

Conclusions. The comparison of calculated wave forms from the FDA to Hammack's experimental results shows that Equations 19 through 24 can be used to estimate wave disturbances from seabed motions in both the generation region and in the far field. The finite-difference approximation has been shown to be stable and accurate for long-wave problems. The FDA can be employed with a variety of bottom motion functions to estimate tsunami forms in one dimension. The general form of Equations 13 through 18 can be employed to formulate more general two-dimensional bottom disturbance models.

Hammack's Experiments: — — —

Numerical Results: — — —

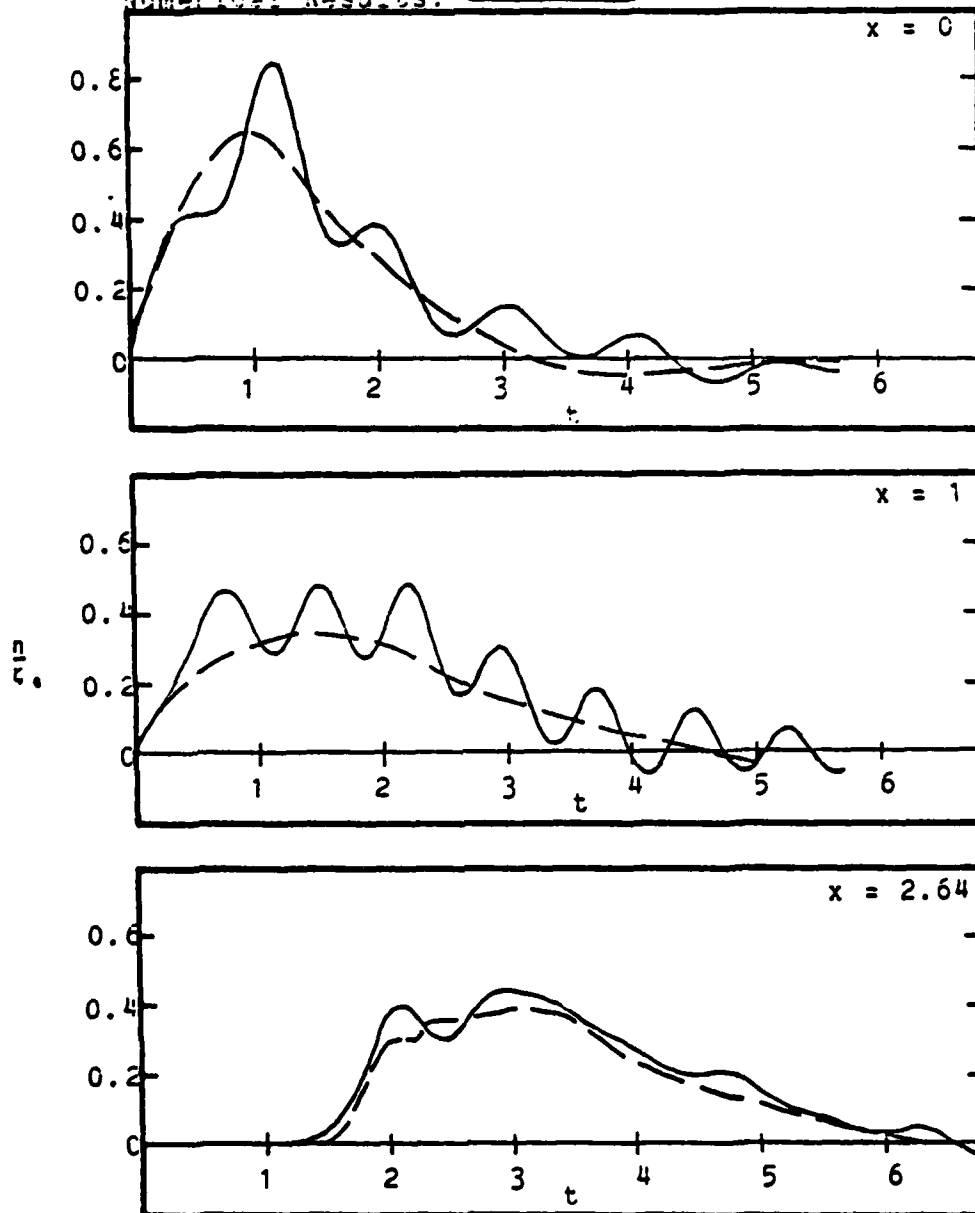


Figure 13
Free-Surface Fluctuations at $x = 0$, $x = 1$,
and $x = 2.64$ vs. Time:
Numerical Calculations vs. Hammack's Data, for
Wave Formation in the Generation Region

$\Delta t = 0.01$ $\zeta_0 = 0.041$
 $\Delta x = 0.1$ $\alpha_0 = 1.5857$
 $\epsilon = 0.0020$ $t_c = 0.70$

Hammack's Experiments: — — — —

Numerical Results: — — — —

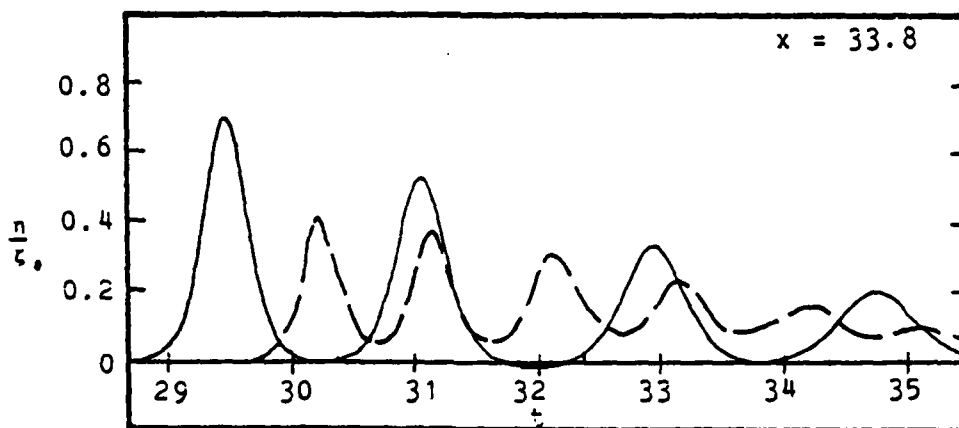
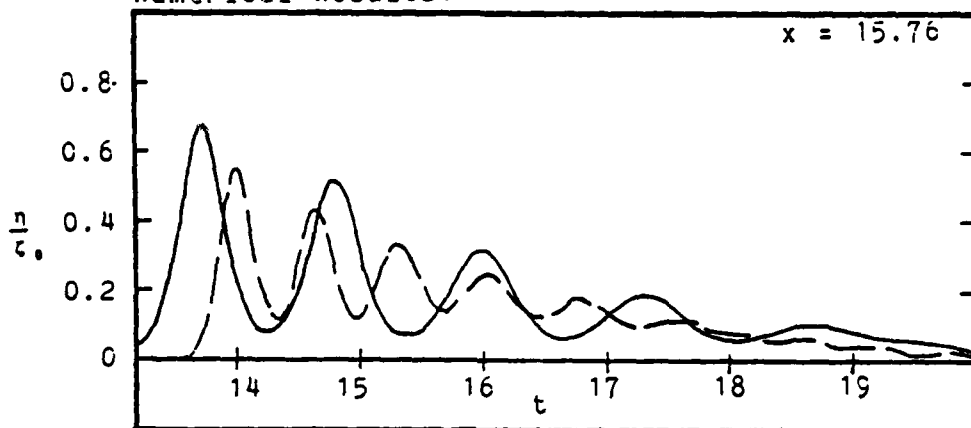


Figure 14
Free-Surface Fluctuations at $x = 15.76$ and
 $x = 33.8$ vs. Time:
Numerical Calculations vs. Hammack's Data, for
Wave Propagation in the Far-Field

$\Delta t = 0.01$	$\tau_0 = 0.041$
$\Delta x = 0.1$	$\alpha_0 = 1.5857$
$\epsilon = 0.0820$	$\tau_c = 0.70$

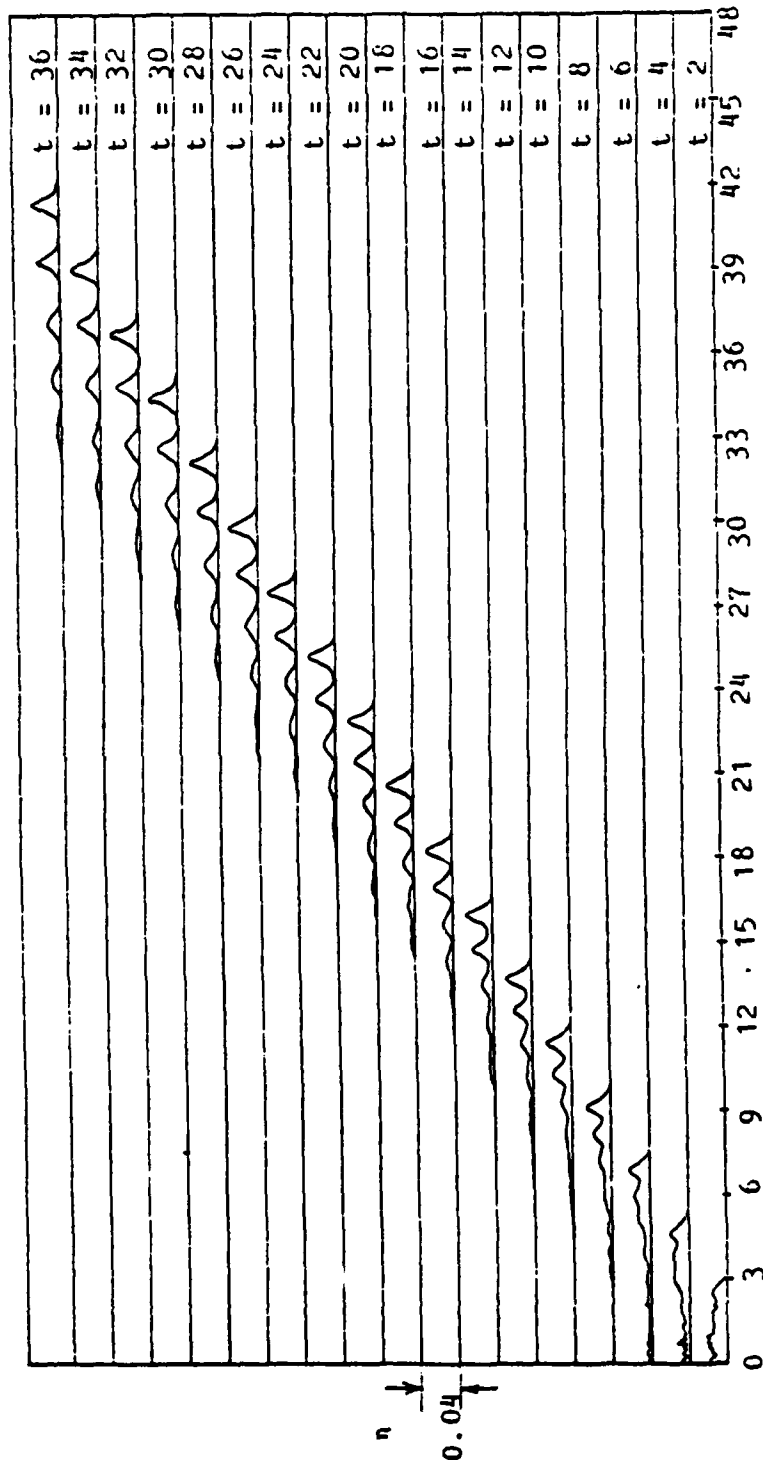


Figure 15
 Calculated Free-Surface Profiles over Time:
 Solitary Wave Formation in the Propagation Region

$$\begin{aligned} \Delta t &= 0.01 & \zeta_0 &= 0.041 \\ \Delta x &= 0.1 & \alpha_0 &= 1.5857 \\ \epsilon &= 0.0820 & t_c &= 0.70 \end{aligned}$$

Acknowledgments. This study was performed under the direction and guidance of Professor Philip L-F Liu at the Civil and Environmental Engineering Department of Cornell University. Funding was provided by the Sea Grant Institute, U. S. Department of Commerce, and the National Science Foundation. All computations were performed on the IBM 370/168 at Cornell.

References

- Boussinesq, M. J. (1872). "Theorie des ondes et des remous qui se propagent le long d'un canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement pareilles de la surface au fond," Journal des Mathematiques Pures et Appliquees, 2nd serie, Vol. 17, pp. 55-108.
- Braddock, R. D., P. Van Den Driessche, and G. W. Peady (1973). "Tsunami Generation," Journal of Fluid Mechanics, Vol. 59, Part 4, pp. 817-828.
- Hammack, J. R. (1972). Tsunamis--A Model of Their Generation and Propagation, California Institute of Technology Report No. KH-R-28 (doctoral thesis).
- Hwang, L. S., and David Divoky (1970). "Tsunami Generation," Journal of Geophysical Research, Vol. 75, No. 33, pp. 6802-6817.
- Hwang, L. S., H. L. Butler, David Divoky (1972). "Tsunami Model: Generation and Open-Sea Characteristics," Bulletin of the Seismological Society of America, Vol. 62, No. 6, pp. 1579-1596.
- Hwang, L. S., and David Divoky (1975). Numerical Investigations of Tsunami Behavior, Tetra-Tech Report, Public Bulletin 257-487.
- Keulegan, G. H. (1948). "Gradual Damping of Solitary Waves," Journal of Research of the National Bureau of Standards, Vol. 40, pp. 487-498.
- Korteweg, D.-J., and G. deVries (1895). "On the Change of Form of Long Waves Advancing in a Rectangular Channel, and on a New Type of Long Stationary Wave," London, Edinburgh, and Dublin Philosophical Magazine, Series 5, Vol. 39, pp. 422-443.
- Laitone, E. V. (1963). "Higher Order Approximations to Nonlinear Waves and the Limiting Heights of Cnoidal, Solitary, and Stokes' Waves," Beach Erosion Board, U. S. Department of the Army, Corps of Engineers, Technical Memorandum No. 133.
- Lin, C. C., and A. Clark (1959). Journal of Chinese Studies, Tsing Hua University, Spec. No. 1, Nat. Sci., 54.
- Madsen, O. S., and C. C. Mei (1969). Dispersive Long Waves of Finite Amplitude over an Uneven Bottom, Massachusetts Institute of Technology Hydrodynamics Laboratory Report No. 117.
- Mei, C. C., and B. LeMehaute (1966). "Notes on the Equations of Long Waves over an Uneven Bottom," Journal of Geophysical Research, Vol. 71, No. 2, pp. 393-400.

Peregrine, D. H. (1966). "Calculation of the Development of an Undular Bore," Journal of Fluid Mechanics, Vol. 25, Part 2, pp. 321-330.

Peregrine, D. H. (1967). "Long Waves on a Beach," Journal of Fluid Mechanics, Vol. 27, Part 4, pp. 815-827.

Sano, K., and K. Hasegawa (1915). "On the Wave Produced by a Sudden Depression of a Small Portion of the Sea Bottom," Bulletin of the Central Meteorological Observatory of Japan, Vol. 2, No. 3, pp. 1-30.

Syono, S. (1936). "On the Wave Caused by a Sudden Deformation of a Finite Portion of the Bottom of a Sea at Uniform Depth," Geophysical Magazine of Japan, Vol. 10, pp. 21-42.

Tuck, E. O., and L. S. Hwang (1972). "Long Wave Generation on a Sloping Beach," Journal of Fluid Mechanics, Vol. 51, Part 3, pp. 449-461.

Ursell, F. (1953). "The Long-Wave Paradox in the Theory of Gravity Waves," Cambridge Philosophical Society, Proceedings, Vol. 49, pp. 685-694.

THEORY AND CALCULATION OF THE NON-LINEAR ENERGY
TRANSFER BETWEEN SEA WAVES IN DEEP WATER

Barbara A. Tracy
U.S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

Donald T. Resio*
Oceanweather, Inc.
Vicksburg, Mississippi 39180

ABSTRACT. Non-linear coupling between sea waves results when the continuity equation is solved using a free surface for boundary conditions. Hasselmann was able to provide a solution to these equations by a perturbation of the linear solution since the non-linear coupling is weak. Hasselmann assumed a Gaussian sea and set up a solution which could be analyzed by looking at the Boltzmann integral for $\frac{dn}{dt}$. This paper uses Webb's technique to solve the energy transfer Boltzmann integrals and discusses how this integration process has been made simpler and more efficient by the utilization of a geometrically-spaced polar grid over the spectral region. This grid allows the loci and the coefficients inside the integrand to scale by various multiples of the geometric scaling factor. Numerical results for the non-linear energy transfer are given for various spectra.

1. INTRODUCTION. The linear processes that produce waves in the ocean are fairly well understood, but the non-linear effects such as wave breaking and non-linear interactions are just beginning to be studied. The sea wave spectrum consists of a peak before a low wavenumber cutoff, and the JONSWAP investigators in the North Sea showed that this peak shifts to a lower wavenumber cutoff as time increases. The non-linear wave-wave interaction could be an explanation for this peak shift. The non-linear interactions transfer energy to different parts of the spectrum and help the sea go back to an equilibrium condition after the wind has added energy to the sea. An evaluation of this non-linear energy transfer is valuable for studies of ocean waves.

2. STATEMENT OF PROBLEM. In order to formulate the problem of the non-linear transfer in physical terms, Hasselmann (1962) treated the waves like particles where each wave reacted like a packet of momentum. He treated the waves in a set of four waves where there were three active participants and one passive participant. The energy transfer problem can then be treated like a coupled mechanical system and can be solved by a perturbation of the linear solution since the non-linear couplings are weak. The evaluation of the non-linear transfer term is done statistically considering that the sea is Gaussian. The process is very similar to the particle scattering problem. This paper takes Hasselmann's theory and Webb's (1978) technique of evaluating the non-linear energy transfer integral and describes how a specialized polar grid with equal angle increments and radials spaced by a geometric progression will simplify evaluation of the integral in terms of computer time and provide physical insight into the interaction process.

* Formerly with U. S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi 39180.

3. THEORY. The theory is based on a fifth order perturbation of the series solutions for the velocity potential and the surface deviation using the non-linear system of equations for the irrotational motion of a horizontally unbounded ideal fluid with finite constant depth and a free surface. Using this method of solution, Hasselmann has written an integral expression for the time rate of change of action density:

$$\frac{dn_1}{dt} = \iiint d\vec{k}_4 d\vec{k}_2 d\vec{k}_3 C(\vec{k}_1, \vec{k}_2, \vec{k}_3, \vec{k}_4) \cdot \delta(\vec{k}_1 + \vec{k}_2 - \vec{k}_3 - \vec{k}_4) \delta(\omega_1 + \omega_2 - \omega_3 - \omega_4) \cdot (n_1 n_3 (n_4 - n_2) + n_2 n_4 (n_3 - n_1)).$$

This integral contains a coupling or interaction coefficient, a density expression and two delta functions - one for conservation of momentum and one for conservation of energy. The set of four waves being considered each have wavenumber, \vec{k}_1 , and angular momentum, ω_1 . Webb (1978) has evaluated this integral by considering the integrand coefficients as a transfer function which evaluates the rate wavenumber \vec{k}_3 is scattered into \vec{k}_1 . The evaluation of the delta functions results in a wavenumber configuration, $\vec{k}_1 + \vec{k}_2 - \vec{k}_3 = \vec{k}_4$, and an expression for the angular velocities, $\omega_1 + \omega_2 - \omega_3 = \omega_4$, which reflects the conservation of energy condition. The integral can be evaluated by considering a specific $\vec{k}_3 - \vec{k}_1$ interaction defined by $\vec{P} = \vec{k}_3 - \vec{k}_1$. In a k_1, k_2, k_3 co-ordinate system one specific intersection of k_3 and k_1 will result in a whole line of applicable k_2 -values that will satisfy the energy conservation conditions. We know that $\omega \propto k^{1/2}$ in deep water so we can write the energy conservation condition in terms of the constant values for k_1 and k_3 and can solve the expression analytically for the k_2 -values. These k_2 -values will be in the form of an egg-shaped locus in a cartesian k_2 -space. By writing the energy conservation condition as a function of k_2 , we can evaluate the integral in the co-ordinate system normal and tangential to the locus and integrate around the closed curve of the locus. The change of co-ordinate systems contributes a gradient of the argument of the energy conservation condition to the integral. Evaluation of the integral now entails evaluation of each possible locus for each possible $\vec{k}_1 - \vec{k}_3$ combination and evaluation of the factors of the integrand at each of these interaction conditions. These integrand factors include the coupling coefficient, the density function and the gradient of the locus function.

4. EVALUATION OF THE INTEGRAL USING A SPECIALIZED GRID. Normally, evaluation of the integral would proceed by considering a regularly spaced polar grid where the k -values will be the radial values and the θ values will be the angles between the interaction k -values. The integration would proceed by considering each possible $\vec{P} = \vec{k}_3 - \vec{k}_1$ vector on the grid and numerically integrating over each interaction by evaluating all the factors of the integrand at each of these interaction conditions. Instead, a special form of a polar grid, see Figure 1, was set up where the radials were spaced in a geometric progression. The first radial value, k_0 , was multiplied by λ to produce the second radial, and the second radial value was multiplied by λ to produce the

third radial, $\lambda^2 k_0$, and so on. As can be seen in Figure 1 a set of parallel interaction \vec{P} -vectors is produced between the 18° and 0° radials. By considering the geometry of these parallel \vec{P} -vectors we find that the ratio of the magnitude of \vec{P}_2 to the magnitude of \vec{P}_1 is equal to λ . If we compare the co-ordinates of the two locus equations for \vec{P}_1 and \vec{P}_2 we find that $x_2 = \lambda x_1$ and $y_2 = \lambda y_1$. Therefore, if we have the co-ordinates for the \vec{P}_1 -locus, we can calculate the co-ordinates of the \vec{P}_2 -locus and the other parallel \vec{P} -vector loci by just multiplying the initial locus co-ordinates by the appropriate multiple of λ . Since the calculation of each separate locus for each interaction would require a lot of computer time, the specialized grid allows us to calculate only a basic set of loci which can be used over and over.

The other factors of the integrand which are dependent on the specific interaction and the specific locus also scale by various factors of λ , the geometric spacing constant. The phase space or gradient term scales by $\lambda^{1/2}$ and the coupling coefficient scales by λ^6 . The ds differential along the locus curve scales by λ . All these factors can be combined into a common scale factor, $\lambda^{15/2}$. Again, a basic set of phase space terms and coupling co-efficients can be set up with the corresponding loci co-ordinates to be used with the appropriate λ value for numerical evaluation of the integral.

The \vec{P} -vector in Figure 1 can also be rotated to different orientations on the grid. In these cases the basic loci can be rotated to the new \vec{P} -vector position and used in the calculations.

The density function is the only factor of the integrand that was not treated by the scaling factors. The density can be considered to be a sum of a pumped transfer and a diffusive transfer (see Figure 2). The subscripts on the n's refer to the density of the corresponding k_1 -vector. The density of each of the wavenumbers is calculated using the JONSWAP spectral form in Figure 2 to calculate the frequency dependent $E(f)$. $E(k)$ is calculated from $E(f)$. A $\cos^2 \theta$ spreading function was used for the test cases.

5. THE RESULTS. Figure 3 shows the contoured results for the Pierson-Moskowitz spectrum. The Pierson-Moskowitz spectrum can be represented by the JONSWAP spectral function in Figure 2 with $\gamma = 1.0$ and $f_m = 0.3 \text{ sec}^{-1}$. Numerical values compare favorably with Webb's contour results for the Pierson-Moskowitz spectrum. The contour results give the value for $\frac{dn}{dt}$, the non-linear energy transfer at the various orientations of k in the spectral region being considered. The magnitude of k is graphed on the x-axis and the angular orientation of the k -vector is graphed on the y-axis.

The remaining figures show the one-dimensional non-linear energy transfer, $S(f)$, as a function of frequency. The one-dimensional non-linear energy transfer, $S(f)$, is calculated by summing the two-dimensional transfer over all the possible angular orientations. The spectral energy function, $E(f)$,

is graphed with a dotted line on top of the non-linear transfer. The γ and σ parameters listed on the titles of the graphs are the shape parameters in the analytical JONSWAP spectral representation in Figure 2. f_m was left equal to 0.3 Hz in all cases. σ_a is the value of σ when f is below f_m and σ_b is the value of σ when f is greater than f_m . All the results show the positive transfer on the low frequency side of the spectrum, the negative transfer in the region around and above f_m , and the positive transfer at the region of high frequency. The spectra with $\gamma = 12.0$ is especially interesting since it seems to oscillate between positive and negative transfer in the high frequency region, and this spectra seems to demonstrate how the non-linear energy transfer is attempting to bring the energy system back to an equilibrium condition by transferring the energy to different parts of the spectrum.

6. CONCLUSION. The specialized grid presented in this paper is an effective time-saving tool when calculating the non-linear energy transfer for a whole spectrum. Routines that have been developed in the past have been too expensive to run for a whole spectrum. The computer time for the Pierson-Moskowitz spectrum using thirty values of k (0.14 m^{-1} to 2.44 m^{-1}) and angular increments from -90° to $+90^\circ$ in 4.5° increments was 151 seconds on the CRAY computer.

If the reader is interested in a more complete discussion of the theory and method discussed in this paper, the same topic will be covered in more depth in Technical Report 11 of the Wave Information Study. This report will be published later this year at the U. S. Army Engineer Waterways Experiment Station.

7. ACKNOWLEDGEMENTS. The U. S. Army Engineer Waterways Experiment Station is acknowledged for supporting the research contained herein. The research and results in this paper were performed as part of the Wave Information Study (WIS) being performed by the U. S. Army Engineer Waterways Experiment Station. This WIS is conducted under the Field Data Collection Program of the U. S. Army Corps of Engineers by the Coastal Engineering Research Center. Permission was granted by the Chief of Engineers to publish this paper.

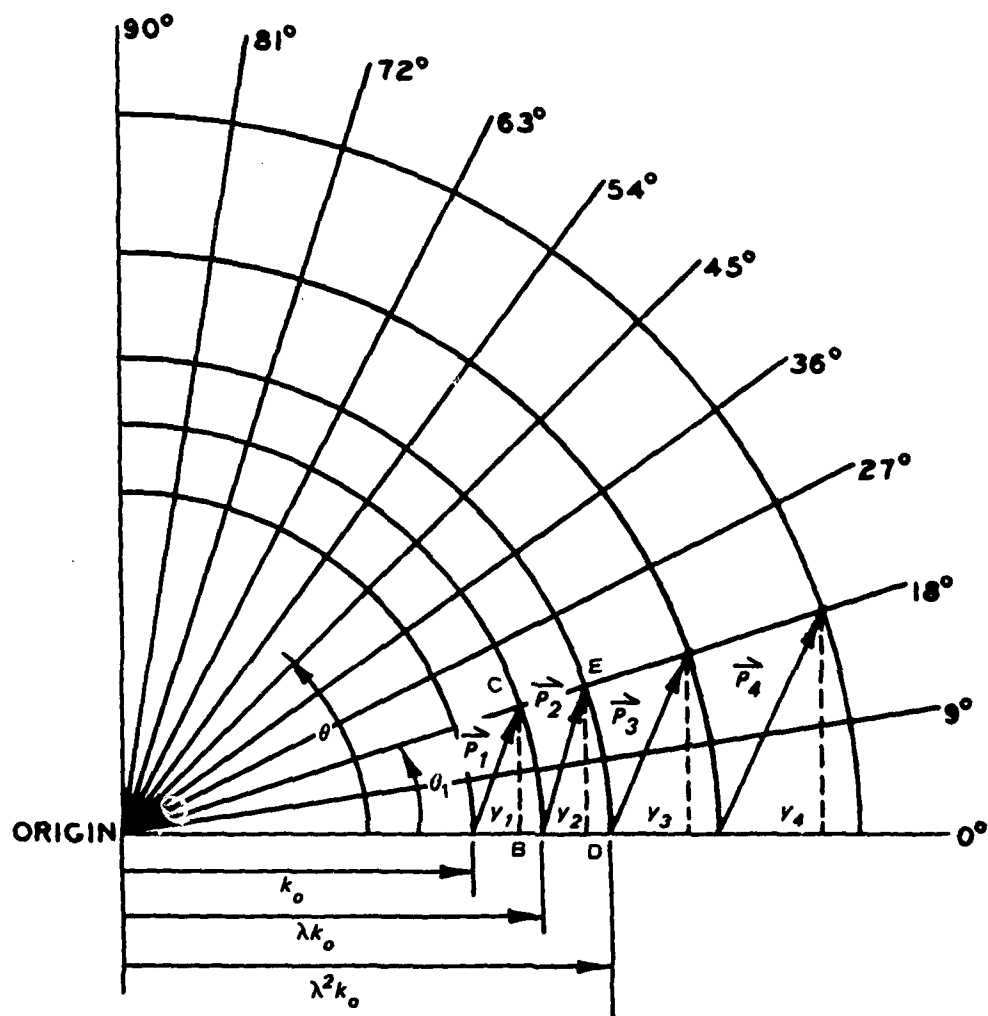


Figure 1. A geometrically spaced polar grid showing a set of parallel interaction (\vec{P}) vectors. k_0 is the smallest radius that was considered and λ is the geometric scaling factor

DENSITY FUNCTION

$$\frac{n_1 n_3 (n_4 - n_2)}{\text{Pumped transfer}} + \frac{n_2 n_4 (n_3 - n_1)}{\text{Diffusive transfer}}$$

$$n_i(\vec{k}_i) = \frac{F(\vec{k}_i)}{\omega_i} \quad \text{where } F(\vec{k}_i) \text{ is two-dimensional spectrum with respect to wave number}$$

$$F(\vec{k}_i) = E(k) \cos^2 \theta$$

where $\cos^2 \theta$ is the spreading function

$E(k)$ is a transformation of $E(f)$ using $c_g = \frac{1}{2} \sqrt{\frac{g}{k}}$

where $E(f)$ is represented using the JONSWAP spectral form

$$E(f) = \alpha g^2 (2\pi)^{-4} f^5 \exp\left(\frac{-5}{4} \left(\frac{f}{f_m}\right)^{-4}\right) \gamma \exp\left(\frac{-(f-f_m)^2}{2\sigma^2 f_m^2}\right)$$

for the Pierson-Moskowitz spectrum:

$$f_m = 0.3$$

$$\alpha = 0.01$$

$$\gamma = 1.0$$

σ is a shape parameter

α is Phillips equilibrium constant

Figure 2. A description of the action density used in Hasselmann's equation and the JONSWAP spectral form used to evaluate the energy at each specific frequency, f , or wavenumber, k . θ is the spreading angle, ω_i is angular velocity, c_g is phase velocity, g is gravity, and f_m is the frequency of the peak of the spectrum

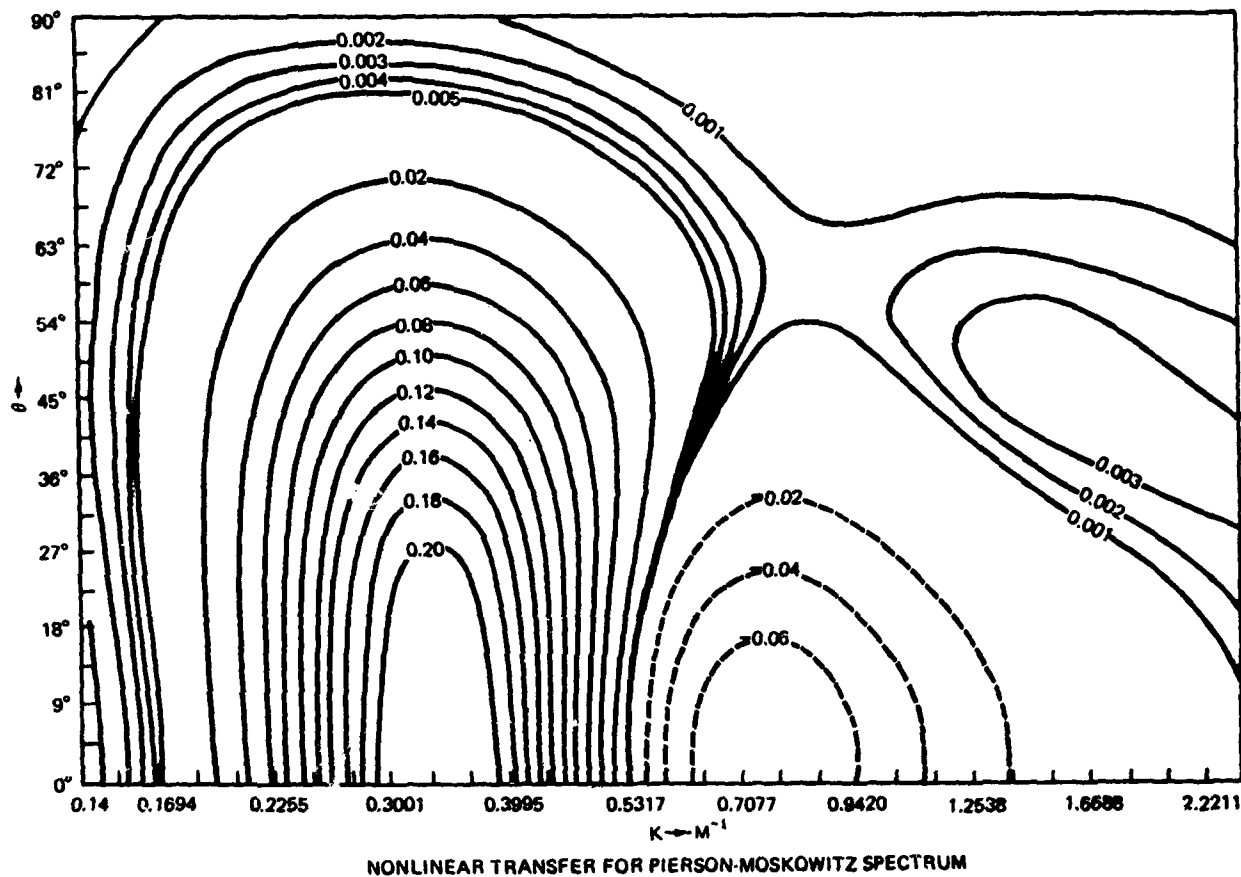
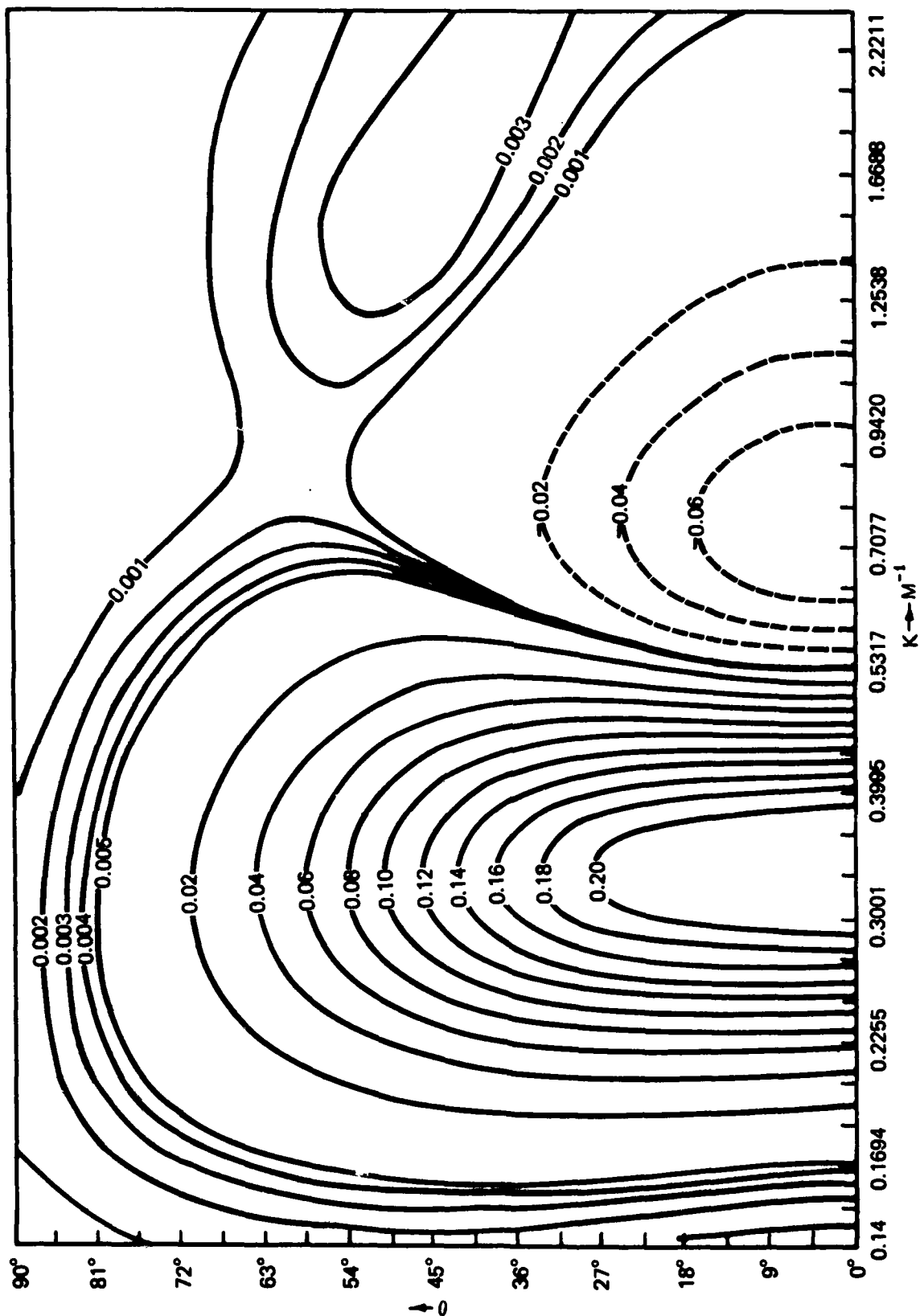


Figure 3. Contoured results of the value of $\frac{dn}{dt}$ for the Pierson-Moskowitz spectrum. Results were obtained in polar co-ordinates (k, θ) where the wavenumber, k , is graphed on the x-axis and the θ value is graphed on the y-axis



NONLINEAR TRANSFER FOR PIERSON-MOSKOWITZ SPECTRUM

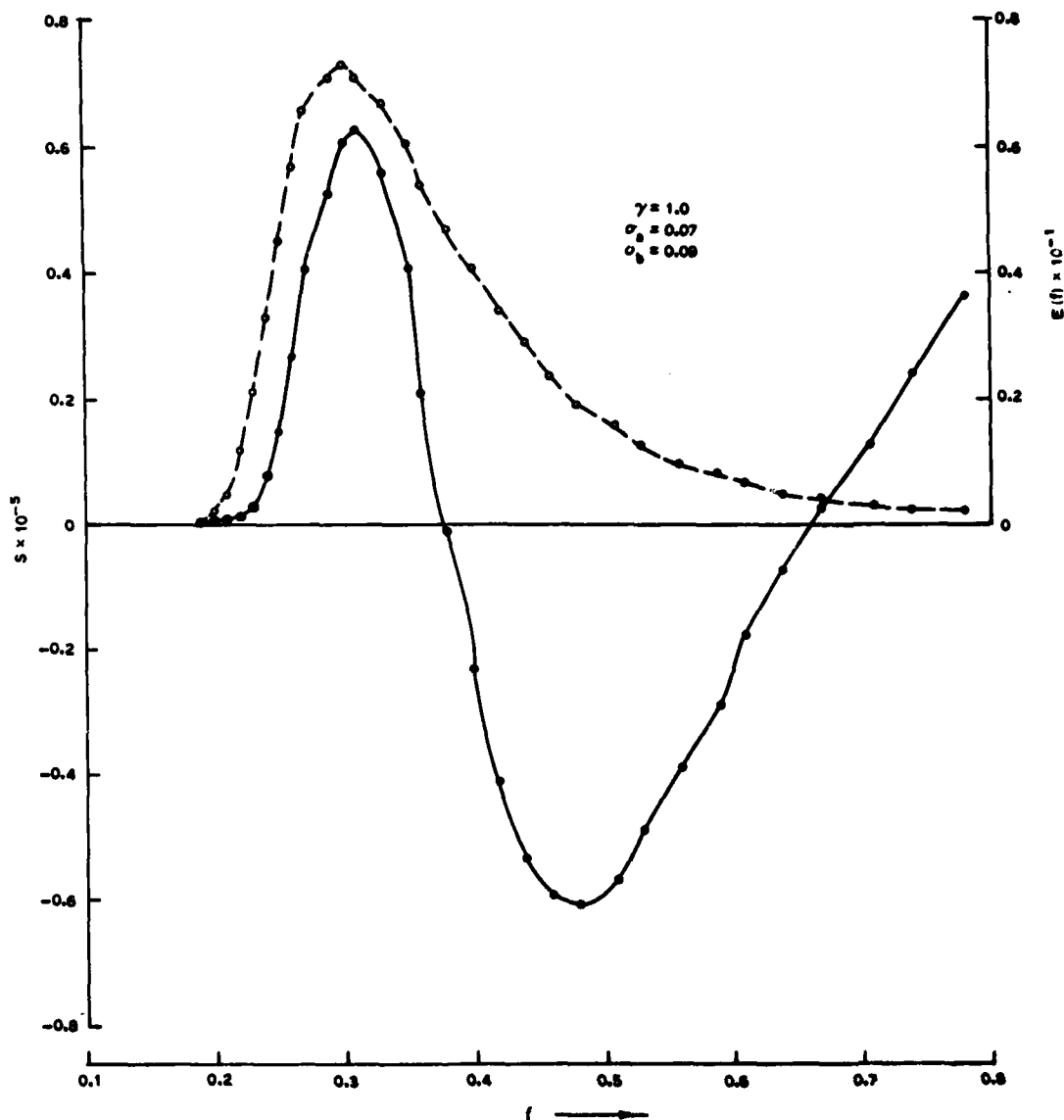


Figure 4. The one-dimensional $\frac{dn}{dt}$ (s) as a function of frequency (solid line) and the energy density of the spectrum ($E(f)$) as a function of frequency (dotted line) for the Pierson-Moskowitz spectrum ($\lambda = 1.0$, $\sigma_a = 0.07$, $\sigma_b = 0.09$)

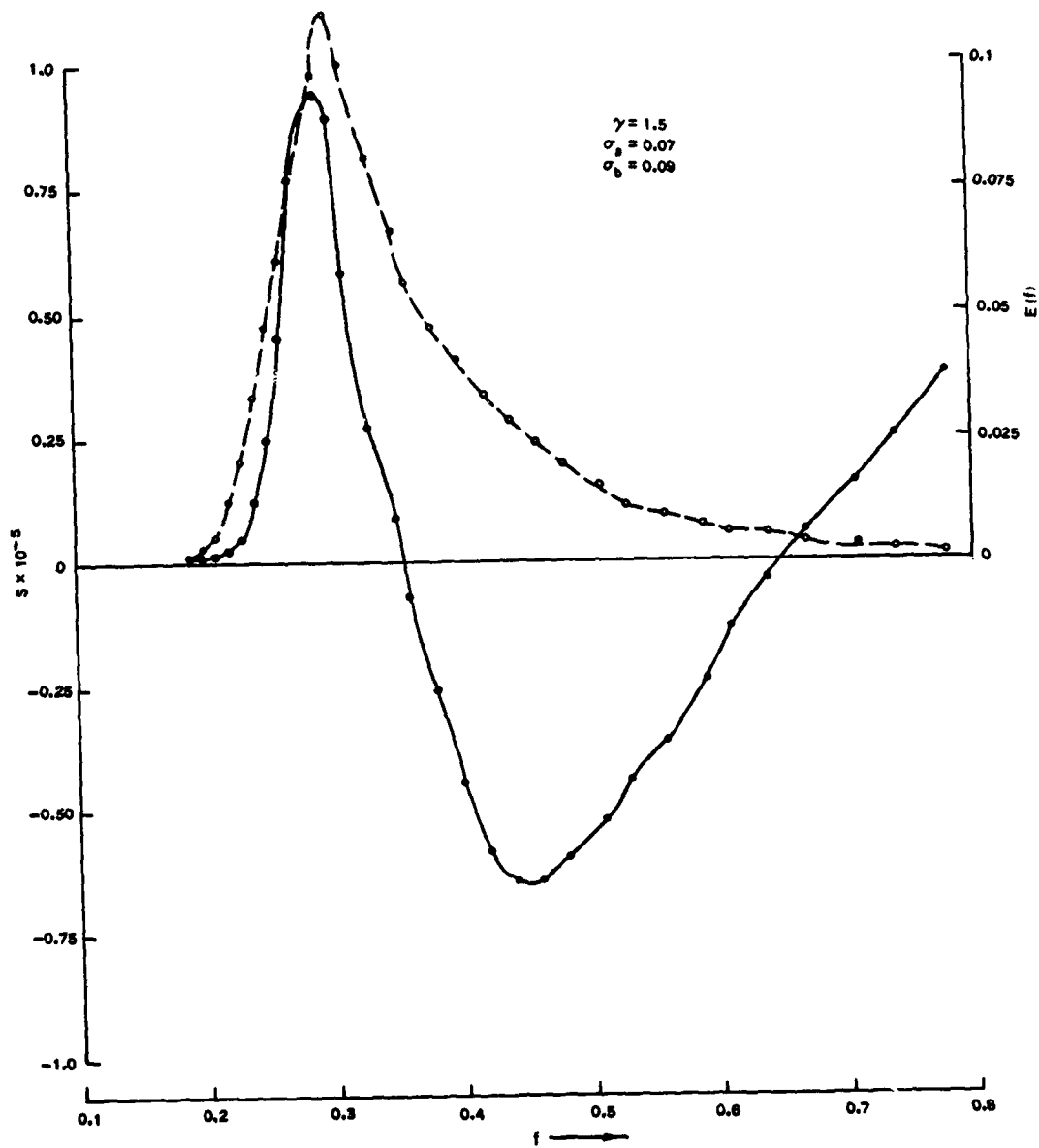


Figure 5. The same type of plot as Figure 4 for spectral parameters as noted on the graph

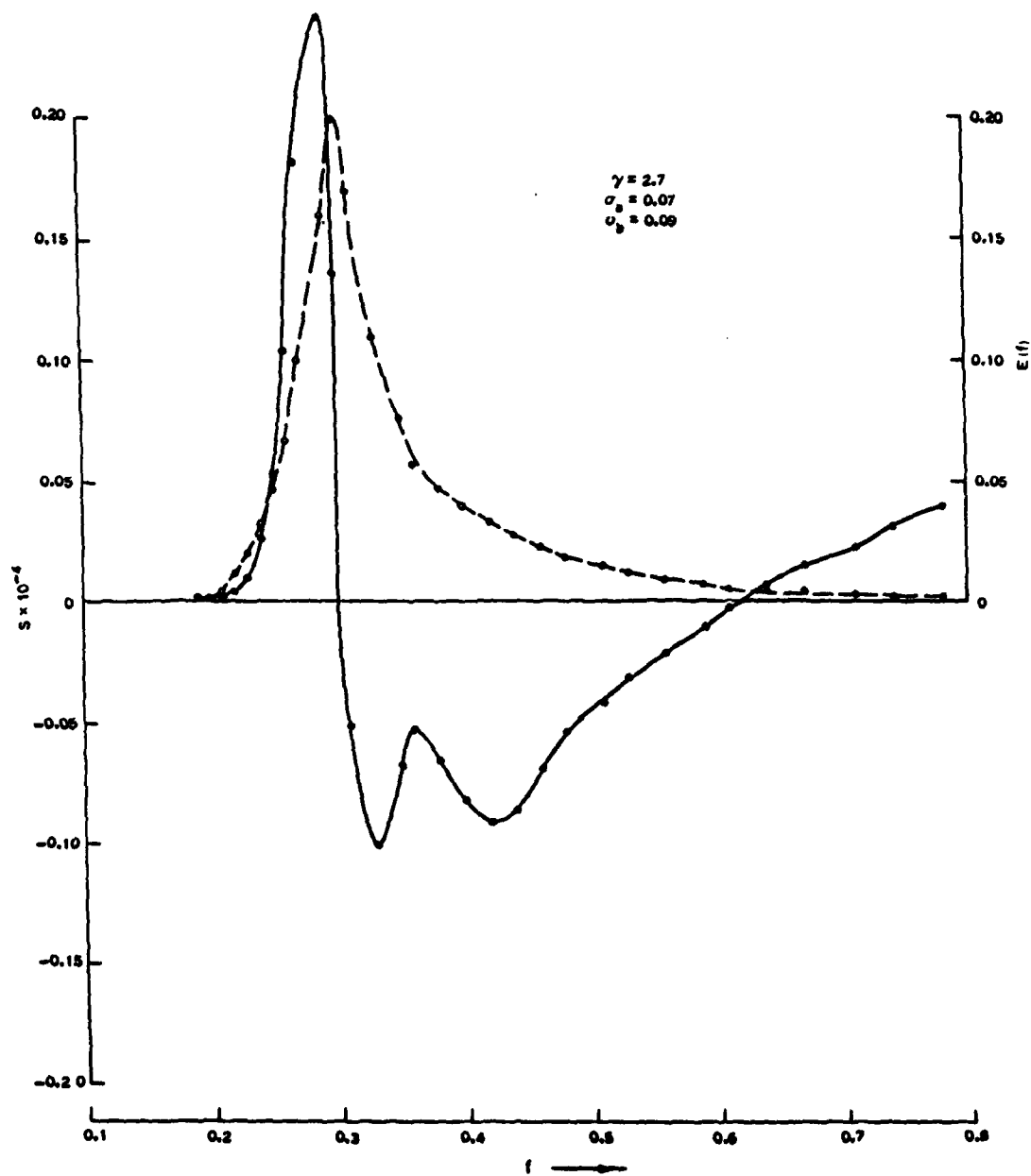


Figure 6. The same type of plot as Figure 4 for spectral parameters as noted on the graph.

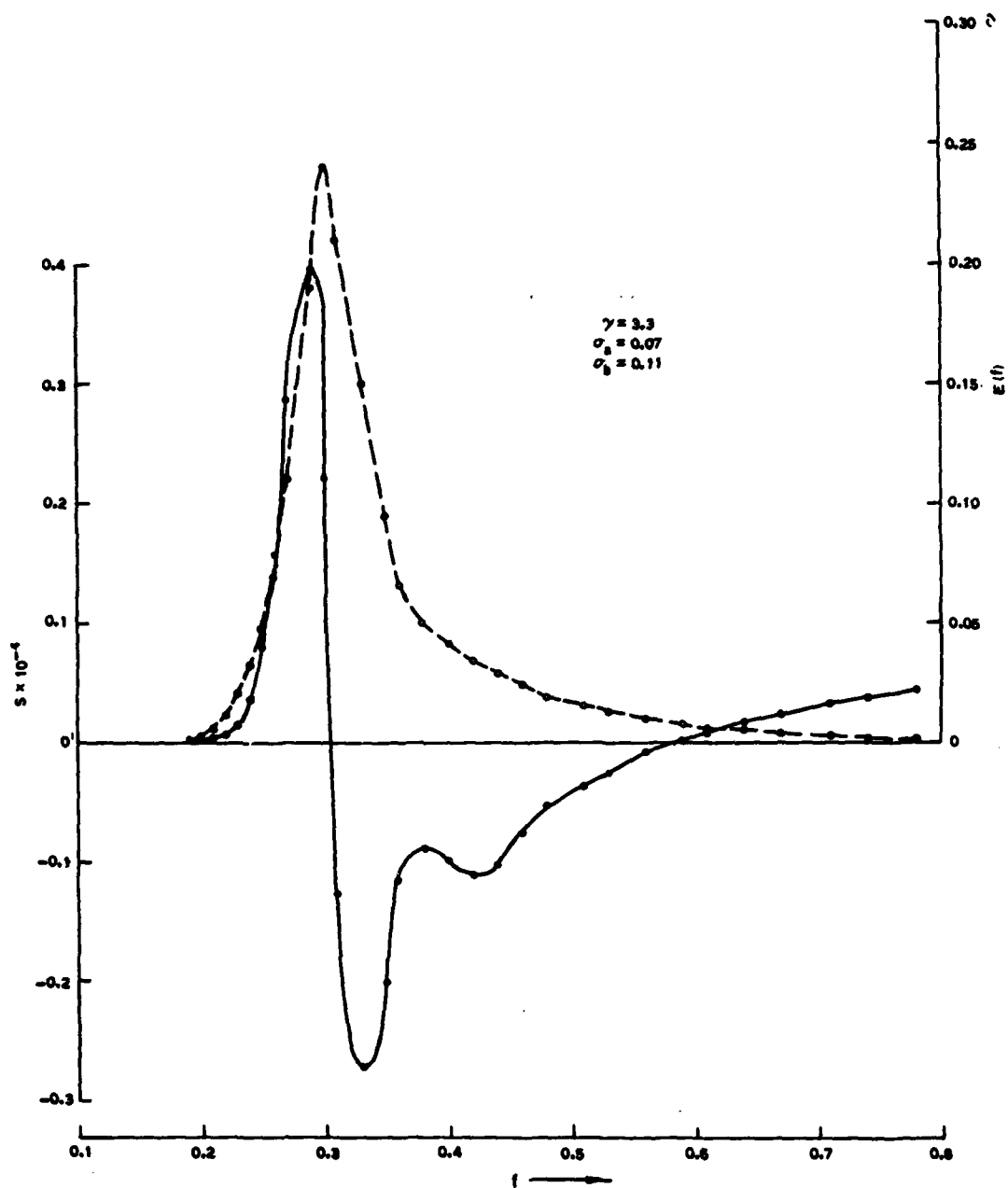


Figure 7. The same type of plot as Figure 4 for spectral parameters as noted on the graph

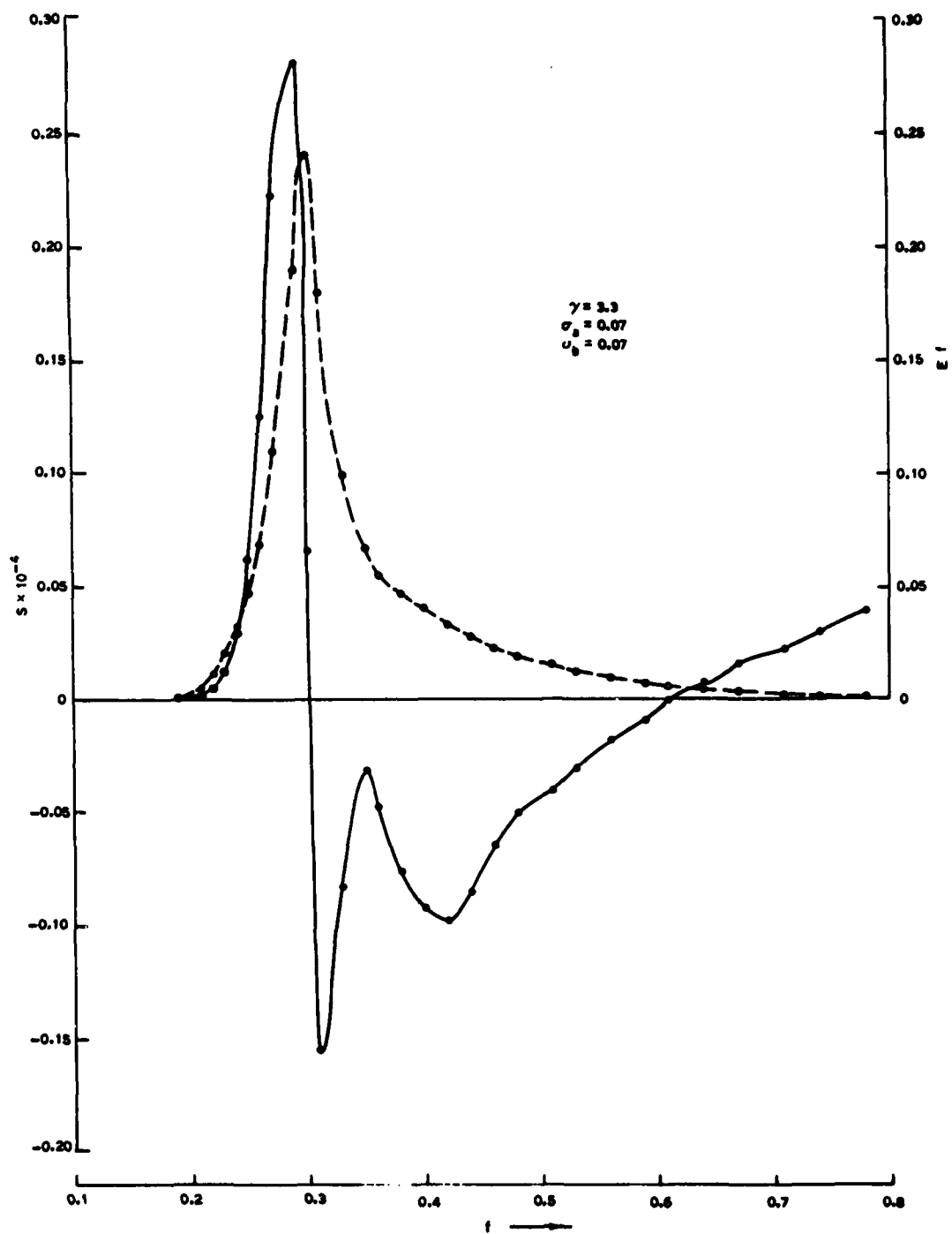


Figure 8. The same type of plot as Figure 4 for spectral parameters as noted on the graph

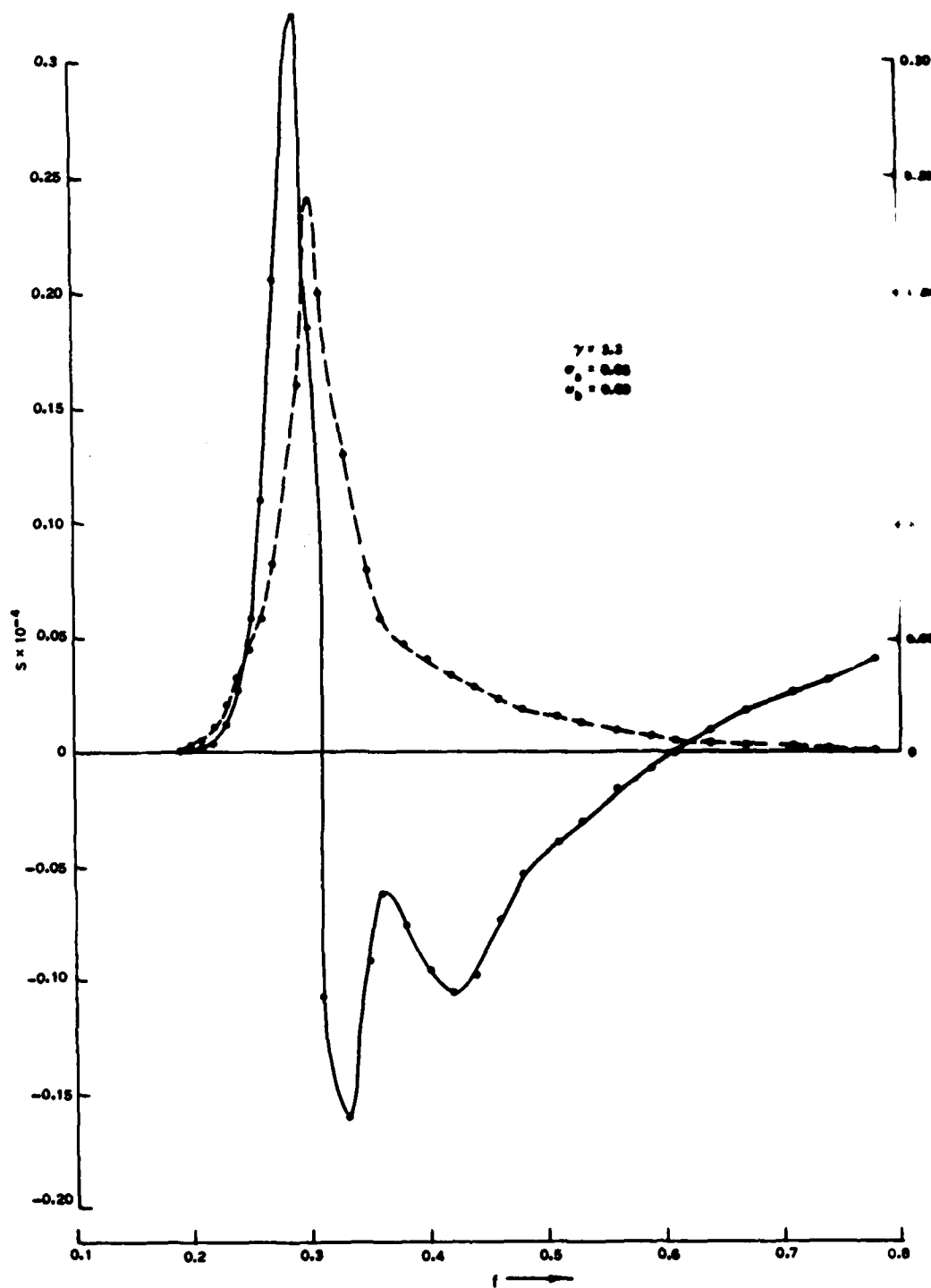


Figure 9. The same type of plot as Figure 4 for spectral parameters as noted on the graph

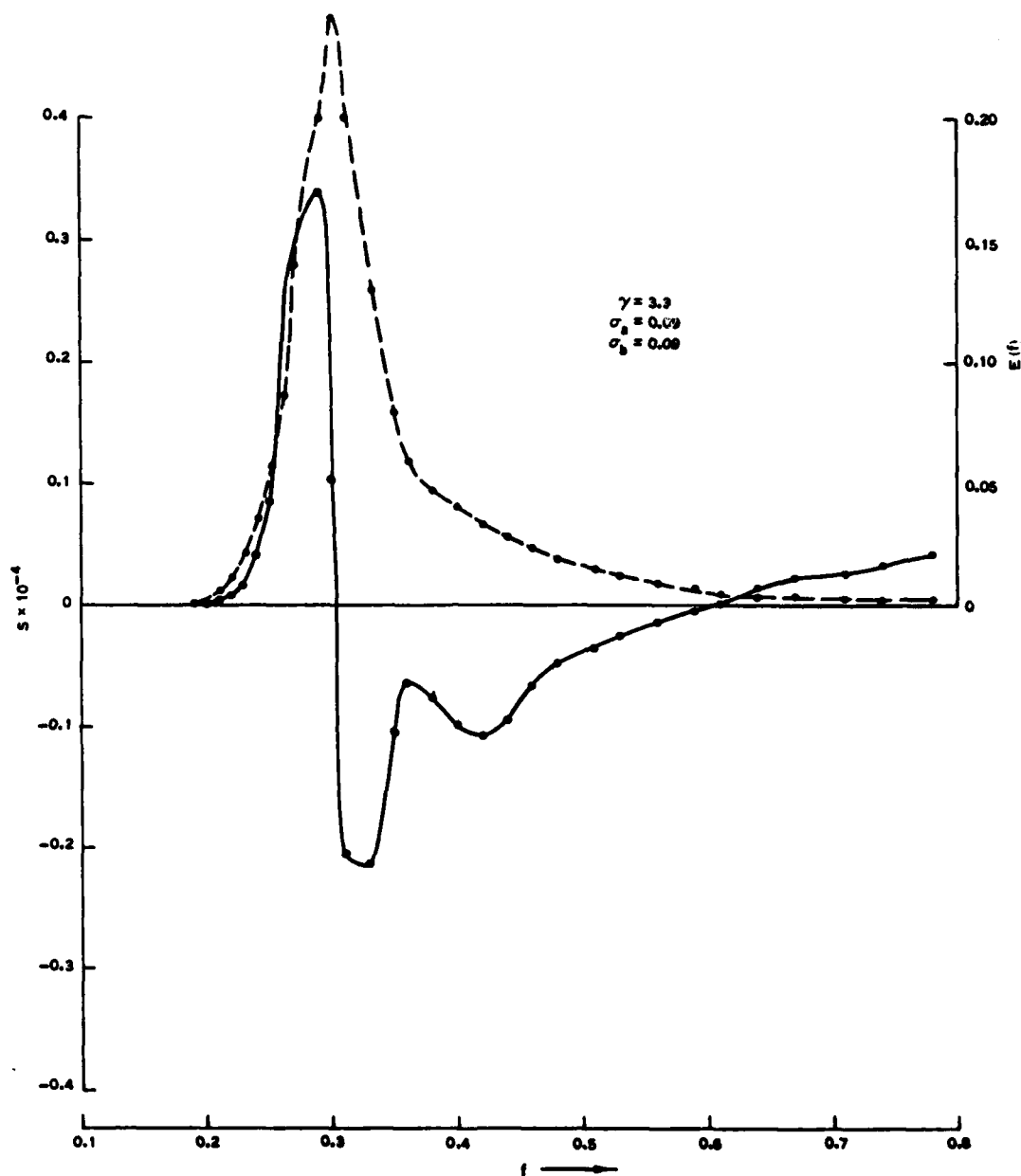


Figure 10. The same type of plot as Figure 4 for spectral parameters as noted on the graph

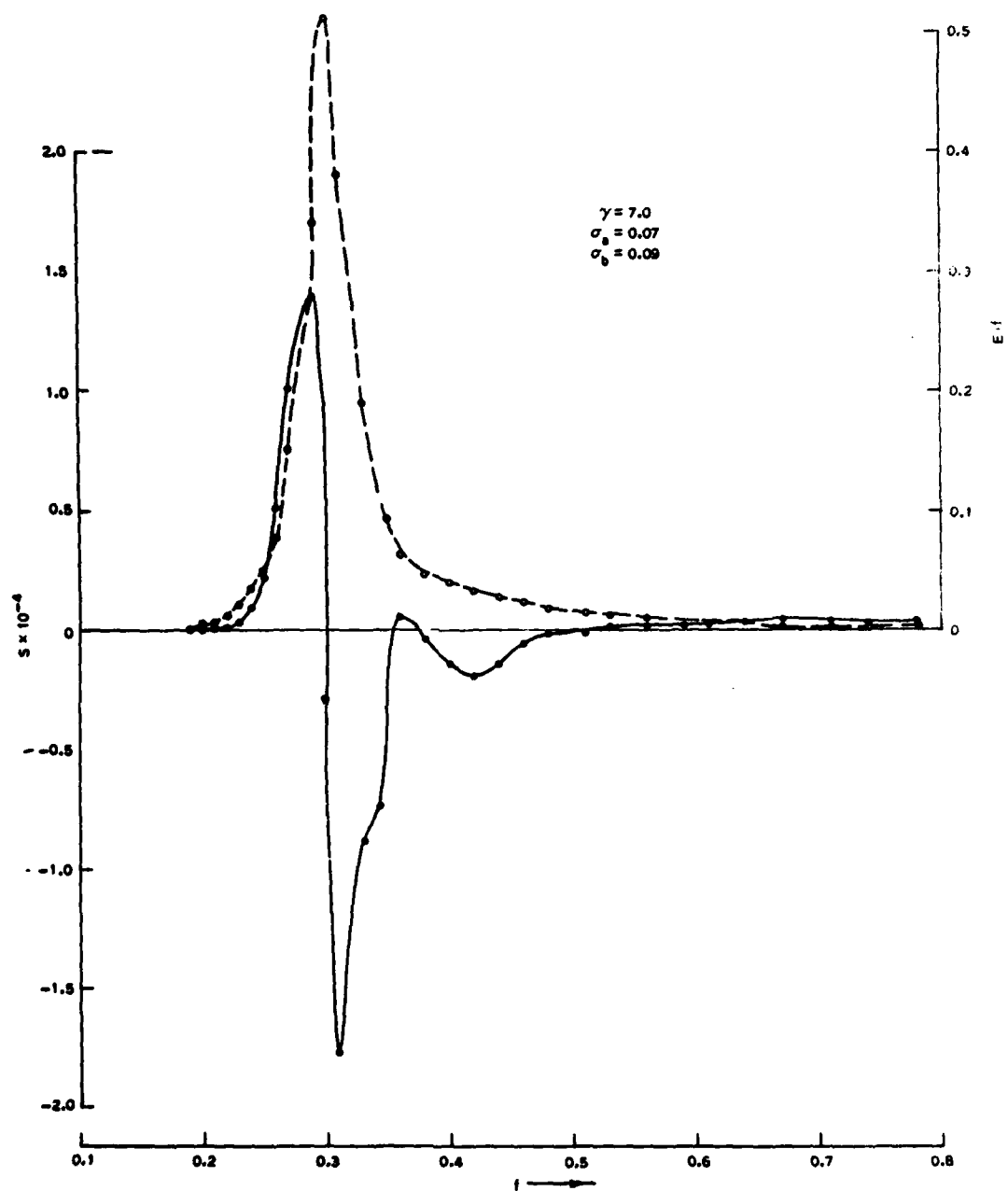


Figure 11. The same type of plot as Figure 4 for spectral parameters as noted on the graph

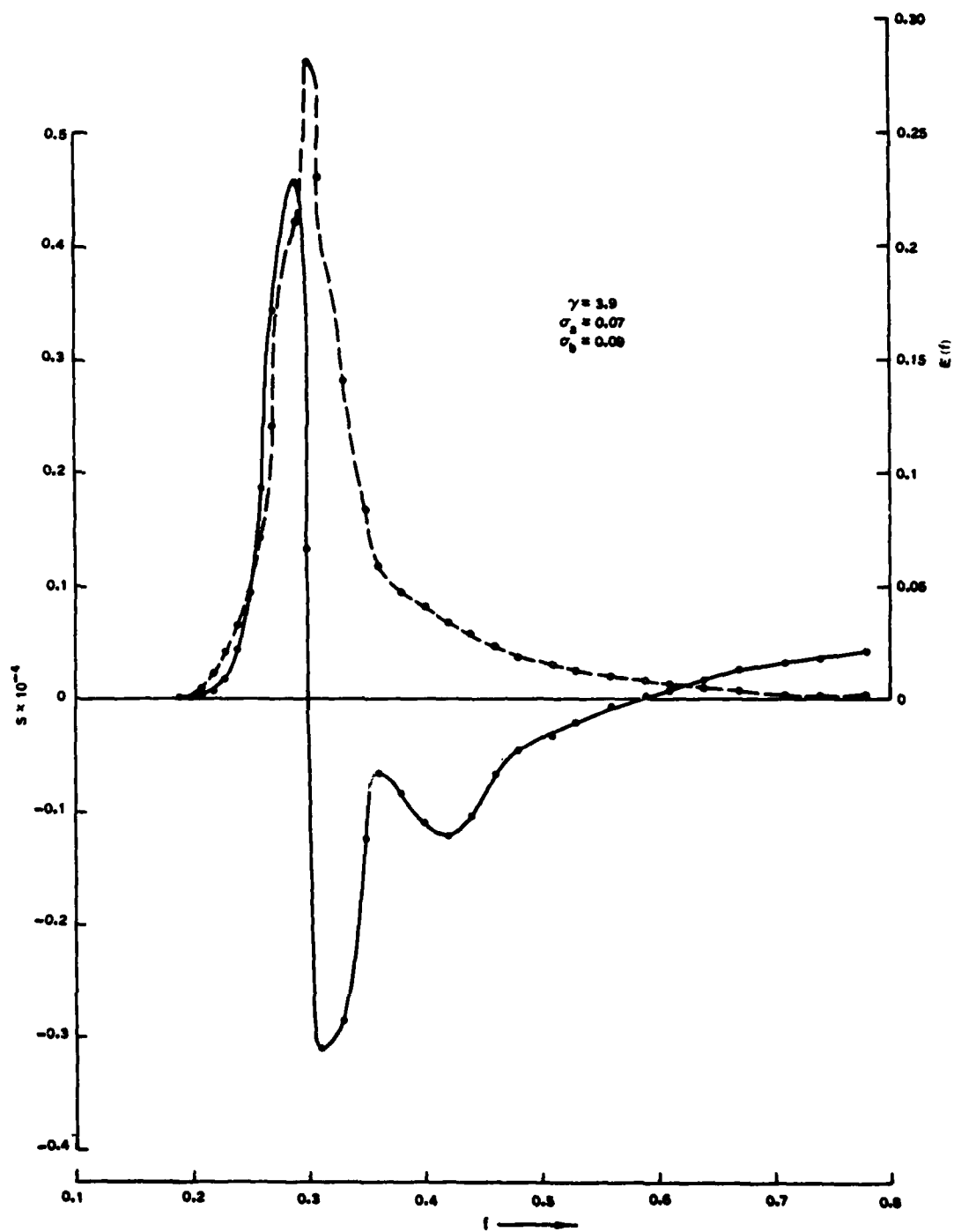


Figure 12. The same type of plot as Figure 4 for spectral parameters as noted on the graph

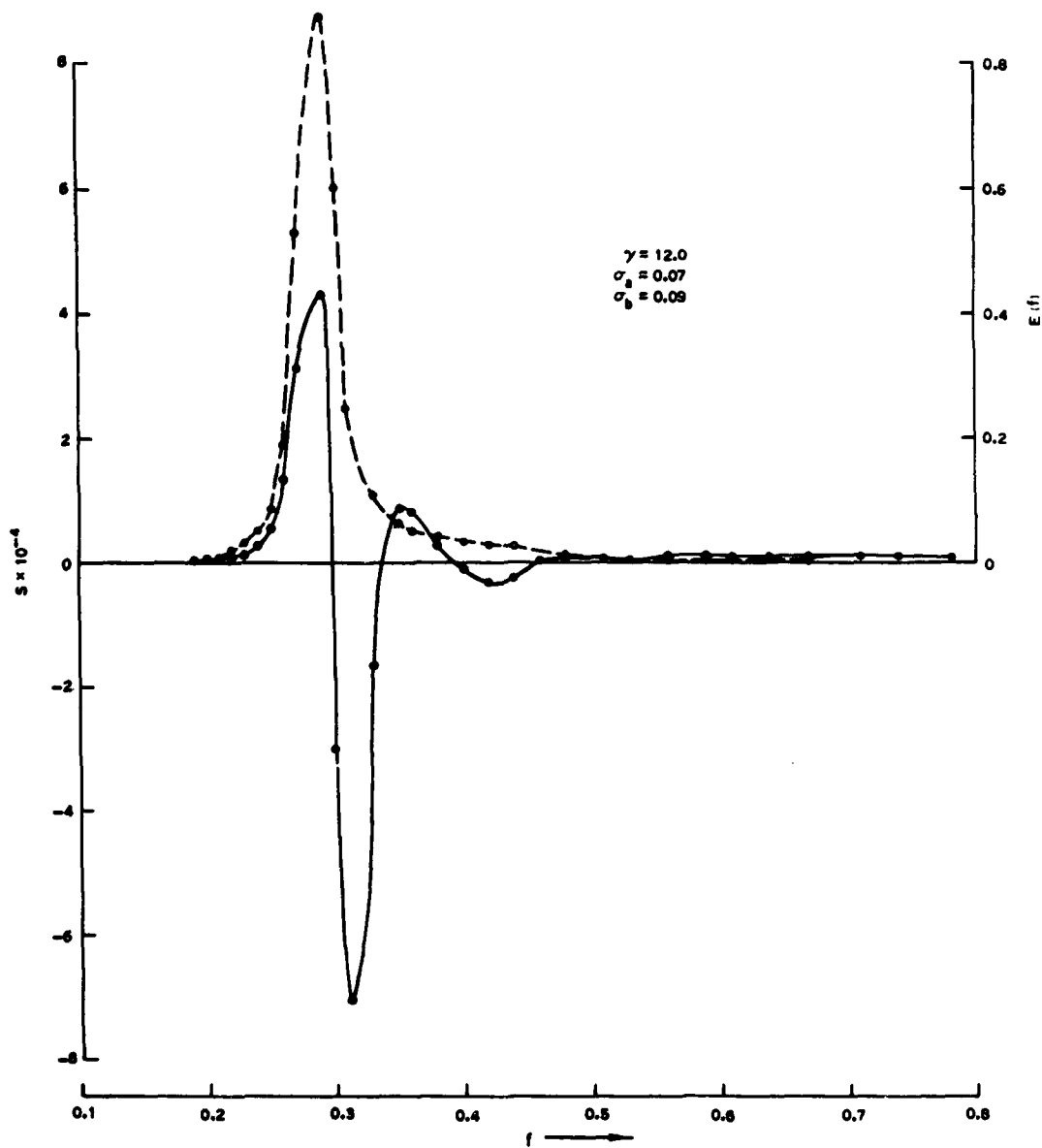


Figure 13. The same type of plot as Figure 4 for spectral parameters as noted on the graph

REFERENCES:

1. Hasselmann, K. (1962). "On the Non-Linear Energy Transfer in Gravity-Wave Spectrum 1: General Theory," Journal of Fluid Mechanics, Vol. 12, pp. 481-500.
2. Jackson, J. D. (1962). Classical Electrodynamics, John Wiley and Sons, Inc., New York.
3. Sell, W., and Hasselmann, K. (1972). "Computations of Non-Linear Energy Transfer for JONSWAP and Empirical Wind Wave Spectra," (Report), Institute of Geophysics, University of Hamburg.
4. Tracy, B. A. (1981). "Calculation of the Non-Linear Energy Transfer Between Sea Waves Using a Geometrically-Spaced Grid," Master's Thesis for Department of Mechanical Engineering, Mississippi State University Mississippi State, MS.
5. Tracy, F. T. (1979). "A Computer Program for Contouring the Output of Finite Element Programs," U. S. Army Engineer Waterways Experiment Station, Automatic Data Processing Center, Vicksburg, MS.
6. Webb, D. J. (1978). "Non-Linear Transfers Between Sea Waves," Deep-Sea Research, Vol. 25, pp. 279-298.

CONSTRAINED AND UNCONSTRAINED VARIATIONAL FINITE ELEMENT FORMULATION
OF SOLUTIONS TO A STRESS WAVE PROBLEM - A NUMERICAL COMPARISON

Julian J. Wu and C. N. Shen
U.S. Army Armament Research & Development Command
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189

ABSTRACT. Unconstrained variational formulation has been applied to initial, boundary value problems previously with some numerical success (refs. 1,2). More recently, an adjoint bilinear variational principle has also been developed for initial and initial-boundary value problems which requires that the initial conditions be satisfied exactly and hence is a constrained variational formulation (refs. 3,4). This present paper compares the numerical results of these two variational formulations for the case of a stress wave problem in a uniform bar.

1. INTRODUCTION. This note presents the solution formulation and finite element discretization of a stress wave problem with discontinuous data in two variational schemes. The first is in a sense a generalized Galerkin's approach in that it works for non-self adjoint problem and that all the end conditions are made to be natural ones and hence none of them are required to be satisfied by the trial functions. This unconstrained variational finite element formulation has been applied to initial/boundary value problems other than wave equations previously (refs. 1,2). More recently, an adjoint bilinear variational principle has been developed for initial and initial/boundary value problems which requires that the initial conditions be satisfied exactly and the variations of the adjoint variable be set to vanish. It is consequently a constrained variational formulation. This note compares one formulation and numerical results with those of the other.

First, in Section 2, the physical problem of a longitudinal stress wave in an elastic rod is stated. The rod is fixed at one end and free at the other end. The discontinuity data arises from the initial linear displacement, corresponding to a constant stress, due to a force applied at the "free" end. This force suddenly disappears at time zero causing a stress discontinuity at the free end. The two variational formulations for the stated problem are introduced in Section 3. Finite element discretization and shape functions are introduced in Section 4. Finally, numerical results and comparisons are made in Section 5.

2. STATEMENT OF THE PROBLEM. The problem considered here is that of a longitudinal stress in a rod. The differential equation can be written as

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{a^2} \frac{\partial^2 u}{\partial t^2} ; \quad \begin{array}{l} 0 < x < l \\ 0 < t < T \end{array} \quad (1)$$

AD-A118 980

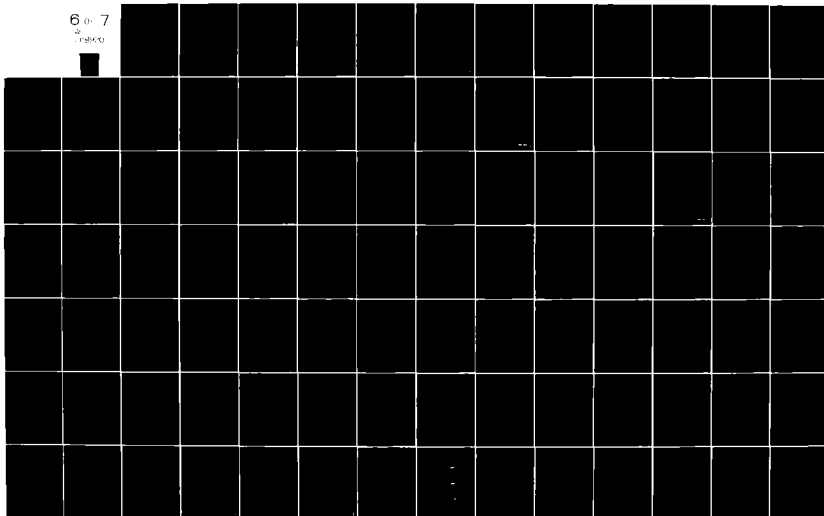
ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82
ARO-82-3

F/8 12/1

UNCLASSIFIED

NL

6 0-7
4
UNCLASSIFIED



with

$$a^2 = E/\rho \quad (2)$$

where $u = u(x,t)$ is the axial displacement
 x,t are the coordinates in axial direction and in time, respectively
 ρ, E are density and Young's modulus, respectively, of the rod material
 l = length of the rod
 T = some finite time of interest

For the boundary conditions, we will consider a rod fixed at one end and not restrained at the other end. Hence

$$\begin{aligned} u(0,t) &= 0 \\ \frac{\partial u}{\partial x}(l,t) &= 0 \end{aligned} \quad (3)$$

The dynamics of the problem is due to the initial conditions. It is assumed that the rod is stretched to a linear displacement by a force P which vanishes at time $t > 0$ (see Figure 1). The initial velocity of the rod is assumed to be zero. Thus

$$\begin{aligned} u(x,0) &= \frac{P}{AE} x \\ \frac{\partial u}{\partial t}(x,0) &= 0 \end{aligned} \quad (4)$$

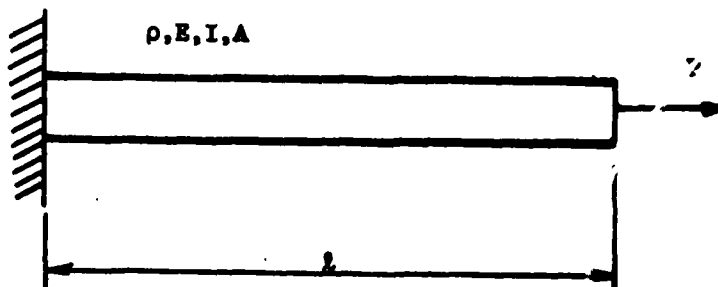


Figure 1. Problem Configuration and Applied Load at Zero Time (i.e., $P = 0$ for $t > 0$).

It is convenient to use dimensionless parameters. Let

$$\bar{u} = u/l, \quad \bar{x} = x/l, \quad \bar{t} = t/T \quad (5)$$

Then, Eq. (1) in dimensionless form is

$$\frac{\partial^2 \bar{u}}{\partial \bar{x}^2} = b^2 \frac{\partial^2 \bar{u}}{\partial \bar{t}^2}, \quad 0 < \bar{x} < 1, \quad 0 < \bar{t} < 1 \quad (6)$$

where

$$b^2 = \frac{1}{a^2} \left(\frac{l}{T}\right)^2 \quad (7)$$

The boundary conditions become

$$\bar{u}(0, \bar{t}) = 0, \quad \frac{\partial \bar{u}}{\partial \bar{x}}(1, \bar{t}) = 0 \quad (8)$$

and

$$\bar{u}(\bar{x}, 0) = P\bar{x}, \quad \frac{\partial \bar{u}}{\partial \bar{t}}(\bar{x}, 0) = 0 \quad (9)$$

where

$$\bar{P} = \frac{P}{AE} \quad (10)$$

is the force in dimensionless form.

The stated problem in dimensionless form are Eqs. (6), (8), and (9) with the new dimensionless parameters related to physical counterparts by Eqs. (5), (6), and (10). To simplify writing, we shall drop the bars in Eqs. (6), (8), and (9), and rewrite them as

$$u'' - b^2 \ddot{u} = 0; \quad 0 < x < 1, \quad 0 < t < 1 \quad (6')$$

$$u(0, t) = 0; \quad u'(1, t) = 0 \quad (8')$$

$$u(x, t) = Px; \quad \dot{u}(x, 0) = 0 \quad (9')$$

where a prime (') indicates differentiation with respect to x and a dot (·), with respect to t .

3. TWO VARIATIONAL FORMULATIONS OF SOLUTIONS. Consider a variational problem

$$\delta I_0 = 0 \quad (11a)$$

with

$$I_0 = I_0(u, v) = \int_0^1 \int_0^1 (-u'v' + b^2 \ddot{u}\ddot{v}) dx dt \quad (11b)$$

where $u(x, t)$ and $v(x, t)$ are said to be adjoint to each other. It is a simple matter to see that this problem is an indeterminate one. However, the functional of Eq. (11b) can be modified to a variational problem which is equivalent to the boundary/initial problem of Eqs. (6'), (8'), and (9'). Thus, consider

$$\delta I = 0 \quad (12a)$$

with

$$\begin{aligned} I = I(u, v) = & \int_0^1 \int_0^1 (-u'v' + b^2 \ddot{u}\ddot{v}) dx dt \\ & + k_1 \int_0^1 u(0, t)v(0, t) dt \\ & + k_2 b^2 \int_0^1 [u(x, 0) - u_0(x)]v(x, 1) dx + b^2 \int_0^1 u_1(x)v(x, 0) dx \end{aligned} \quad (12b)$$

We shall take the first variation of the function $I(u, v)$ of Eq. (12b) in such a manner that δv is completely arbitrary while δu is set to zero identically. Hence, by means of integrations-by-parts, one has

$$\begin{aligned} (\delta I)_{\delta u=0} = & \int_0^1 \int_0^1 (u'' - b^2 \ddot{u}) \delta v dx dt \\ & - \int_0^1 u'(1, t) \delta v(1, t) dt \\ & + \int_0^1 [u'(0, t) + k_1 u(0, t)] \delta v(0, t) dt \\ & + b^2 \int_0^1 \{\dot{u}(x, 1) + k_2 [u(x, 0) - u_0(x)]\} \delta v(x, 1) dx \\ & - b^2 \int_0^1 [\dot{u}(x, 0) - u_1(x)] \delta v(x, 0) dx = 0 \end{aligned} \quad (13)$$

The fact that $\delta v(x,t)$ is completely arbitrary enables us to conclude from Eq. (13) that

$$u'' - b^2 \ddot{u} = 0 \quad ; \quad \begin{array}{l} 0 < x < 1 \\ 0 < t < 1 \end{array}$$

$$u'(1,t) = 0$$

$$u'(0,t) + k_1 u(0,t) = 0 \quad (14)$$

$$\dot{u}(x,1) + k_2 [u(x,0) - u_0(x)] = 0$$

and

$$\dot{u}(x,0) - u_1(x) = 0$$

It is then observed that the initial/boundary value problem defined by Eq. (14) reduces to that of Eqs. (6'), (8'), and (9') if one lets k_1 and k_2 go to infinity (and with $u_0(x) = Px$ and $u_1(x) = 0$). This fact suggests that the variational problem of Eqs. (12) can be used as a basis of a finite element discretization for the approximate solutions to the original initial/boundary problem. It should be noted that all the auxiliary conditions in Eqs. (14) are the so called natural boundary conditions. They are the consequence of the variational problem - just like the differential equation itself. For this reason, the above solution is referred to as an unconstrained variational formulation.

Another approach begins from Eq. (11b). With $\delta u = 0$ once again, one has

$$\begin{aligned} \delta I_0 = & \int_0^1 \int_0^1 (-u' \delta v' + b^2 \dot{u} \delta \dot{v}) dx dt \\ & + \int_0^1 u'(1,t) \delta v(1,t) dt - \int_0^1 u'(0,t) \delta v(0,t) dt \\ & - b^2 \int_0^1 \dot{u}(x,1) \delta v(x,1) dx + b^2 \int_0^1 \dot{u}(x,0) \delta v(x,0) dx = 0 \end{aligned} \quad (15)$$

with the constrained conditions

$$\begin{aligned} u(0,t) = 0 \quad ; \quad u'(1,t) = 0 \quad \text{for } 0 < t < 1 \\ u(x,0) = u_0(x) \quad ; \quad \dot{u}(x,0) = 0 \quad \text{for } 0 < x < 1 \end{aligned} \quad (16)$$

It was shown in another paper (ref. 4) that the variations of the adjoint variable must be constrained as follows

$$\begin{aligned} \delta v(1,t) = 0 \quad ; \quad \delta v'(0,t) = 0 \quad \text{for } 0 < t < 1 \\ \delta v(x,1) = 0 \quad ; \quad \delta \dot{v}(x,1) = 0 \quad \text{for } 0 < x < 1 \end{aligned} \quad (17)$$

4. FINITE ELEMENT DISCRETIZATION AND SHAPE FUNCTIONS. Through nondimensionalization, the region of interest always remains to be a unit square: $0 < x < 1$ and $0 < t < 1$. The finite element discretization is a subdivision of this unit square into smaller rectangles, the elements. A typical element scheme is shown in Figure 2 where a typical (i,j) th element is also shown. In terms of the element variables Eq. (12a) is now written as

$$\delta I = \sum_{i,j} \delta I_{(i,j)} = 0 \quad (18)$$

Variables $u(x,t)$ and $v(x,t)$ become $u_{(i,j)}(\xi,\eta)$ and $v_{(i,j)}(\xi,\eta)$ respectively where ξ,η are local independent variables in spatial and temporal axis also shown in Figure 2.

Relations between global and local coordinates are given as follows

$$\begin{aligned} \xi &= \xi(i) = Kx - i + 1 \\ \eta &= \eta(j) = Lt - j + 1 \end{aligned} \quad (19)$$

where K and L are the number of segments in x and t directions, respectively (see Figure 2).

Shape functions are introduced as follows. Let

$$u_{(i,j)}(\xi,\eta) = \underline{a}^T(\xi,\eta) \underline{U}_{(i,j)} \quad (20)$$

where $\underline{a}(\xi,\eta)$ is the shape function vector and $\underline{U}_{(i,j)}$ is the discretized unknown vector. In this paper, $\underline{a}(\xi,\eta)$ is selected as the following.

Let $a_k(\xi,\eta)$ be a component of vector $\underline{a}(\xi,\eta)$ $k = 1, 2, \dots, 16$, and

$$a_k(\xi,\eta) = b_1(\xi)b_j(\eta) \quad ; \quad \begin{aligned} k &= 1, 2, \dots, 16 \\ i, j &= 1, 2, 3, 4 \end{aligned} \quad (21)$$

with

$$\begin{aligned} b_1(\xi) &= 1 - 3\xi^2 + 2\xi^3 \\ b_2(\xi) &= \xi - 2\xi^2 + \xi^3 \\ b_3(\xi) &= 3\xi^2 - 2\xi^3 \\ b_4(\xi) &= -\xi^2 + \xi^3 \end{aligned} \quad (22)$$

The correspondence between the index k and the pair (i,j) in Eq. (20) is given in Table I.

TABLE I. CORRESPONDENCE BETWEEN k AND (i,j) IN EQ. (20)

k	(i,j)	k	(i,j)
1	(1,1)	9	(1,3)
2	(2,1)	10	(2,3)
3	(1,2)	11	(1,4)
4	(2,2)	12	(2,4)
5	(3,1)	13	(3,3)
6	(4,1)	14	(4,3)
7	(3,2)	15	(3,4)
8	(4,2)	16	(4,4)

With the conventions as stated above, the meaning of the unknowns $U_k(i,j)$ in the vector $U(i,j)$ is as follows

$$\begin{aligned}
 U_1 &= u(0,0) ; U_2 = \frac{\partial u}{\partial \xi}(0,0) ; U_3 = \frac{\partial u}{\partial \eta}(0,0) ; U_4 = \frac{\partial^2 u}{\partial \xi \partial \eta}(0,0) \\
 U_5 &= u(1,0) ; U_6 = \frac{\partial u}{\partial \xi}(1,0) ; U_7 = \frac{\partial u}{\partial \eta}(1,0) ; U_8 = \frac{\partial^2 u}{\partial \xi \partial \eta}(1,0) \\
 U_9 &= u(0,1) ; U_{10} = \frac{\partial u}{\partial \xi}(0,1) ; U_{11} = \frac{\partial u}{\partial \eta}(0,1) ; U_{12} = \frac{\partial^2 u}{\partial \xi \partial \eta}(0,1) \\
 U_{13} &= u(1,1) ; U_{14} = \frac{\partial u}{\partial \xi}(1,1) ; U_{15} = \frac{\partial u}{\partial \eta}(1,1) ; U_{16} = \frac{\partial^2 u}{\partial \xi \partial \eta}(1,1)
 \end{aligned} \quad (23)$$

for each element (i,j) .

For the unconstrained formulation, Eq. (20) is used in Eq. (12). The result is

$$\begin{aligned}
 \sum_{i=1}^K \sum_{j=1}^L \delta \underline{V}(i,j)^T \left\{ -\frac{K}{L} \underline{A} + b^2 \frac{L}{K} \underline{B} \right\} \underline{U}(i,j) + \sum_{j=1}^L \delta \underline{V}(i,j)^T \left(\frac{k_1}{L} \right) \underline{C} \underline{U}(i,j) \\
 + \sum_{i=1}^L \delta \underline{V}(i,1)^T \left(\frac{k_2 b^2}{K} \right) \underline{D} \underline{U}(i,1) = \sum_{i=1}^K \delta \underline{V}(i,L)^T \left(\frac{b^2 k_2}{K} \right) \underline{E}(i) \\
 \sum_{i=1}^K \delta \underline{V}(i,1)^T \left(\frac{b^2}{K} \right) \underline{G}(i)
 \end{aligned} \quad (24)$$

where

$$\begin{aligned} \underline{A} &= \int_0^1 \int_0^1 \underline{a}_{,\xi} \underline{a}^T_{,\xi} d\xi d\eta ; \quad \underline{B} = \int_0^1 \int_0^1 \underline{a}_{,\eta} \underline{a}^T_{,\eta} d\xi d\eta \\ \underline{C} &= \int_0^1 \underline{a}(0,\eta) \underline{a}^T(0,\eta) d\eta ; \quad \underline{D} = \int_0^1 \underline{a}(\xi,1) \underline{a}^T(\xi,0) d\eta \\ \underline{F}(1) &= \int_0^1 u_0(1) \underline{a}(\xi,1) d\xi ; \quad \underline{G}(1) = \int_0^1 u_1(1) \underline{a}(\xi,0) d\xi \end{aligned} \quad (25)$$

The expression of $\underline{F}(1)$ and $\underline{G}(1)$ can be further reduced into a form more readily computed. Write

$$\begin{aligned} u_0(1)(\xi) &= \underline{a}^T(\xi,0) \underline{U}_0(1) = \sum_{k=1}^{16} a_k(\xi,0) U_{0k}(1) \\ u_1(1)(\xi) &= \underline{a}^T_{,\eta}(\xi,0) \underline{U}_0(1) = \sum_{k=1}^{16} a_{k,\eta}(\xi,0) U_{0k}(1) \end{aligned} \quad (26)$$

Since

$$\begin{aligned} a_k(\xi,0) &= b_1(\xi) b_j(0) \\ a_{k,\eta}(\xi,0) &= b_1(\xi) b'_j(0) \end{aligned}$$

and

$$\begin{aligned} b_1(0) &= 1, \quad b_j(0) = 0 \quad \text{for } j = 2,3,4 \\ b'_2(0) &= 1, \quad b'_j(0) = 0 \quad \text{for } j = 1,3,4 \end{aligned}$$

From Table I, one then observes that

$$\begin{aligned} a_k(\xi,0) &= 0 \quad \text{for all } k \text{ except } k = 1,2,5,6 \\ a_{k,\eta}(\xi,0) &= 0 \quad \text{for all } k \text{ except } k = 3,4,7,8 \end{aligned}$$

Hence, in Eq. (26), only $U_{01}(1)$, $U_{02}(1)$, $U_{05}(1)$, and $U_{06}(1)$ are used in expressing $u_0(1)$ and only $U_{03}(1)$, $U_{04}(1)$, $U_{07}(1)$, and $U_{08}(1)$ are used in expressing $u_1(1)(\xi)$. Thus we shall write

$$\begin{aligned} \underline{F}(1) &= \int_0^1 \underline{a}(\xi,1) \underline{a}^T(\xi,0) d\xi \underline{U}_0(1) = \underline{F} \underline{U}_0(1) \\ \underline{G}(1) &= \int_0^1 \underline{a}(\xi,0) \underline{a}^T_{,\eta}(\xi,0) d\xi \underline{U}_0(1) = \underline{G} \underline{U}_0(1) \end{aligned} \quad (27)$$

with

$$\underline{F} = \int_0^1 \underline{a}(\xi, 0) \underline{a}^T(\xi, 0) d\xi \quad (28)$$

$$\underline{G} = \int_0^1 \underline{a}(\xi, 0) \underline{a}^T_{,\eta}(\xi, 0) d\xi$$

The way to set up $\underline{U}_0^{(1)}$ is that first set all $\underline{U}_{0k}^{(1)}$ to zero for all $k = 1, 2, \dots, 16$. That set $\underline{U}_{0k}^{(1)}$ for $k = 1, 2, 3, 4, 5, 6, 7$ and 8 as follows.

$$\begin{aligned} \underline{U}_{01}^{(1)} &= \underline{u}_0^{(1)}(0) ; \quad \underline{U}_{02}^{(1)} = \underline{u}_{0,\xi}^{(1)}(0) ; \quad \underline{U}_{03}^{(1)} = \underline{u}_1^{(1)}(0) ; \\ \underline{U}_{04}^{(1)} &= \underline{u}_{1,\xi}(0) ; \quad \underline{U}_{05} = \underline{u}_0^{(1)}(1) ; \quad \underline{U}_{06}^{(1)} = \underline{u}_{0,\xi}^{(1)}(1) ; \\ \underline{U}_{07}^{(1)} &= \underline{u}_1^{(1)}(1) ; \quad \underline{U}_{08}^{(1)} = \underline{u}_{1,\xi}^{(1)}(1) \end{aligned} \quad (29)$$

With vectors $\underline{U}_0^{(1)}$, \underline{F} , and \underline{G} completely defined above, Eq. (24) can be rewritten as

$$\begin{aligned} & \sum_{i=1}^K \sum_{j=1}^L \delta \underline{V}(i, j)^T \left\{ -\frac{K}{L} \underline{A} + b^2 \frac{L}{K} \underline{B} \right\} \underline{U}(i, j) + \sum_{j=1}^L \delta \underline{V}(i, j)^T \left(\frac{k_1}{L} \underline{C} \right) \underline{U}(i, j) \\ & + \sum_{i=1}^K \delta \underline{V}(i, j)^T \left(\frac{b^2 k_2}{K} \underline{D} \right) \underline{U}(i, 1) = \sum_{i=1}^K \left[\delta \underline{V}(i, L)^T \left(\frac{b^2 k_2}{K} \right) \underline{F} - \delta \underline{V}(i, 1)^T \left(\frac{b^2}{K} \right) \underline{G} \right] \underline{U}_0^{(1)} \end{aligned} \quad (30)$$

Now Eq. (30) is readily assembled into a global matrix equation in a standard manner.

$$\delta \underline{V}^T \underline{K} \underline{U} = \delta \underline{V}^T \underline{P} \quad (31)$$

or

$$\underline{K} \underline{U} = \underline{P} \quad (31)$$

due to the fact $\delta \underline{V}$ is completely arbitrary. Thus Eq. (31) is solved for \underline{U} .

5. NUMERICAL RESULTS AND COMPARISONS. Some preliminary results of computation are presented here. We shall set $b = 1$ in the differential equation (6') for simplicity. Thus,

$$b^2 = \frac{\rho}{E} \frac{\ell^2}{T^2} = \frac{\ell^2}{a^2 T^2} = 1 \quad (32)$$

or,

$$T = \frac{l}{a} \quad (33)$$

The exact solution for $t = 0, 0.2T$, and $0.4T$ are given in Figures 3 and 4.

First, the results from the unconstrained variational formulation. Using a grid scheme of 5×1 , the numerical results for displacements and axial stresses are tabulated in Tables II and III where the exact solutions are also given for comparison. The graphic comparisons are shown in Figures 5 and 6 where the calculated solutions are indicated by crosses (x) and the exact solution is plotted in solid lines. It is clear in these figures that the computed results generally agree with the exact analytical solution. As a further evidence of convergence, a finer grid scheme of 10×1 is taken and the improved solution is shown in Figures 7 and 8.

Numerical results from the constrained formulation follow the general trend as the unconstrained one as indicated in Figures 9 and 10, as well as the tabulated comparison with the exact solution in Tables IV and V.

TABLE II. SOLUTIONS TO THE STRESS WAVE PROBLEM USING UNCONSTRAINED VARIATIONAL FORMULATION

$$t = 0.2T = 0.2\left(\frac{l}{a}\right), b = 1.0; \text{Grid: } 5 \times 1$$

x	u		$\partial u / \partial x$	
	Computed	Exact	Computed	Exact
0	0.000	0.0	0.994	1.0
0.2	0.199	0.2	0.989	1.0
0.4	0.399	0.4	0.965	1.0
0.6	0.598	0.6	0.896	1.0
0.8	0.789	0.8*	0.550	0.0*
1.0	0.806	0.8	0.403	0.0

*Point of discontinuity.

TABLE III. SOLUTIONS TO THE STRESS WAVE PROBLEM USING
UNCONSTRAINED VARIATIONAL FORMULATION

$$t = 0.4T = 0.4\left(\frac{b}{a}\right), b = 1.0; \text{Grid: } 5 \times 1$$

x	u		$\partial u / \partial x$	
	Computed	Exact	Computed	Exact
0	0.000	0.0	0.988	1.0
0.2	0.200	0.2	0.976	1.0
0.4	0.399	0.4	0.927	1.0
0.6	0.594	0.6	0.714	0.0*
0.8	0.698	0.6	0.467	0.0
1.0	0.791	0.6	0.151	0.0

*Point of discontinuity.

TABLE IV. SOLUTIONS TO THE STRESS WAVE PROBLEM USING
CONSTRAINED VARIATIONAL FORMULATION

$$t = 0.1T, b = 1; \text{Grid: } 5 \times 1$$

x	u		$\partial u / \partial x$	
	Computed	Exact	Computed	Exact
0	0.0	0.0	0.986	1.0
0.2	0.200	0.2	0.984	1.0
0.4	0.399	0.4	0.944	1.0
0.6	0.596	0.6	0.784	1.0
0.8	0.797	0.8*	- 0.049	0.0*
1.0	0.874	0.8	0.0	0.0

*Point of discontinuity.

TABLE V. SOLUTIONS TO THE STRESS WAVE PROBLEM USING
CONSTRAINED VARIATIONAL FORMULATION

$t = 0.1T$, $b = 1.0$; Grid: 5×1

x	u		$\partial u / \partial x$	
	Computed	Exact	Computed	Exact
0	0.	0.0	1.000	1.0
0.1	0.100	0.1	1.000	1.0
0.2	0.200	0.2	1.000	1.0
0.2	0.300	0.3	1.000	1.0
0.4	0.400	0.4	0.999	1.0
0.5	0.500	0.5	0.997	1.0
0.6	0.600	0.6	0.986	1.0
0.7	0.700	0.7	0.945	1.0
0.8	0.798	0.8	0.787	1.0
0.9	0.898	0.9*	- 0.038	0.0*
1.0	0.936	0.9	0.0	0.0

*Point of discontinuity.

REFERENCES

1. J. J. Wu, "The Initial Boundary Value of Gun Dynamics Solved by Finite Element Unconstrained Variational Formulations," Innovative Numerical Analysis For the Applied Engineering Science, R. P. Shaw, et al, Editors, University Press of Virginia, Charlottesville, pp. 733-741, 1980.
2. J. J. Wu, "Solutions to Initial Value Problems by Use of Finite-Elements-Unconstrained Variational Formulations," 1977 Journal of Sound and Vibration, 53, pp. 341-356.
3. C. N. Shen and J. J. Wu, "A New Variational Method for Initial Value Problems, Using Piecewise Hermite Polynomial Spline Functions," presented at the 1981 Army Numerical Analysis & Computers Conference, Huntsville, AL, February 1981.
4. C. N. Shen, "Method of Solution for Variational Principle Using Bicubic Hermite Polynomial," presented at the 17th Conference of Army Mathematicians, West Point, NY, June 1981.

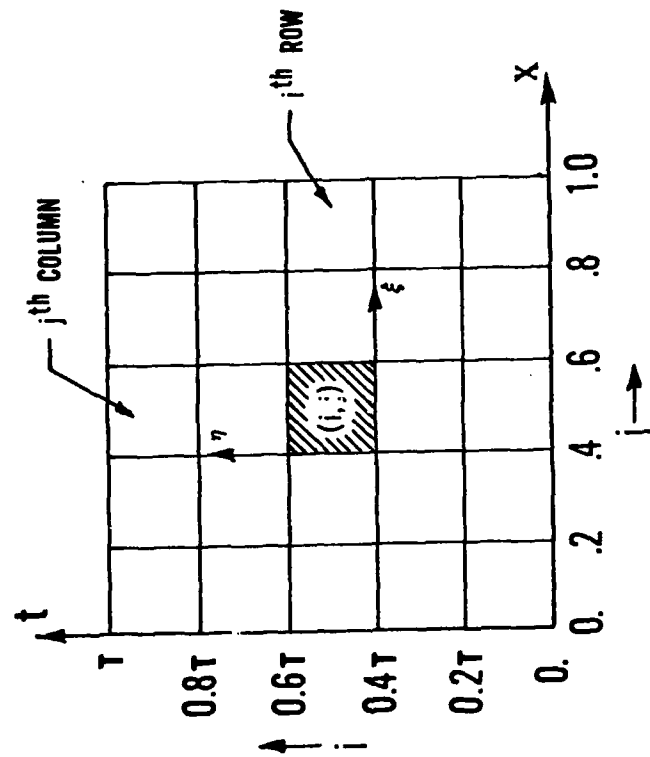


Figure 2. A Typical Finite Element Grid Scheme Showing the (i,j) th Element and the Global, Local Coordinates.

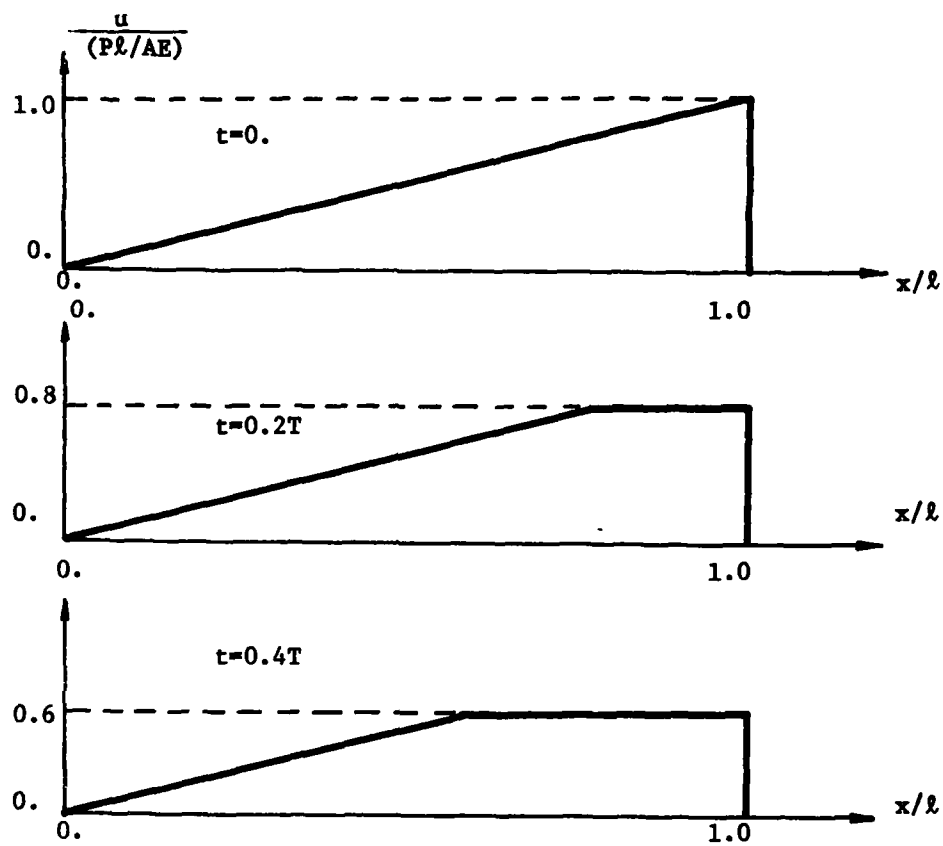


Figure 3. Exact Solution to the Problem: Longitudinal Displacement at $t = 0, 0.2T$, and $0.4T$.

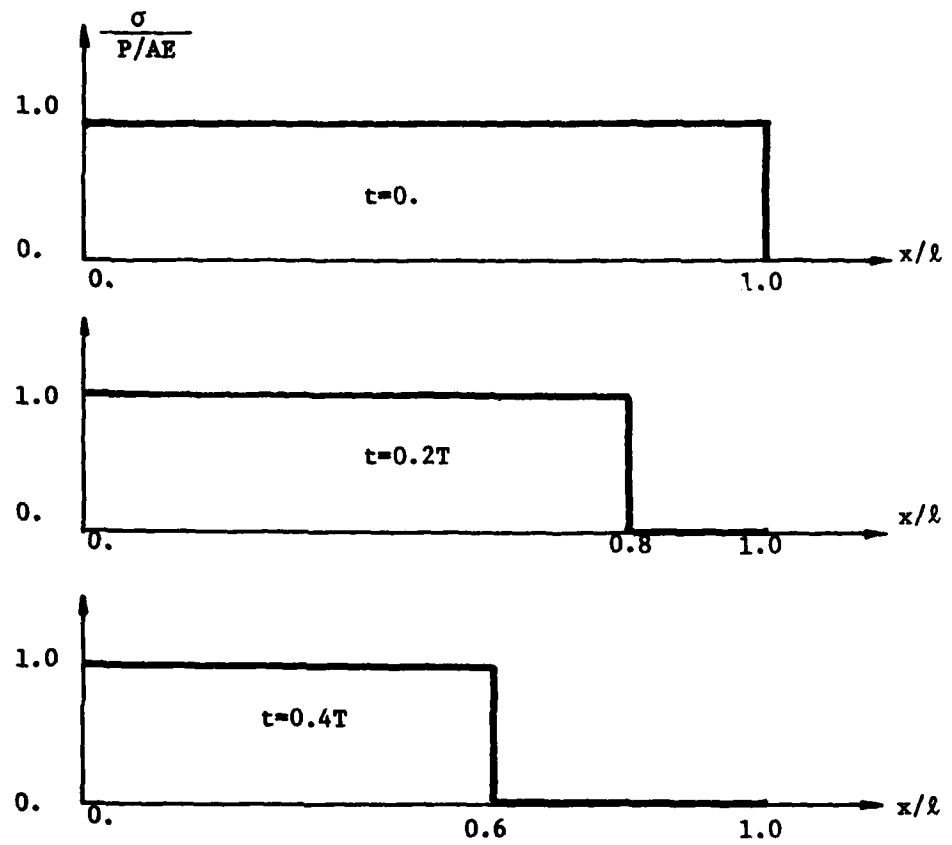


Figure 4. Exact Solution to the Problem: Axial Stress at $t = 0, 0.2T$, and $0.4T$.

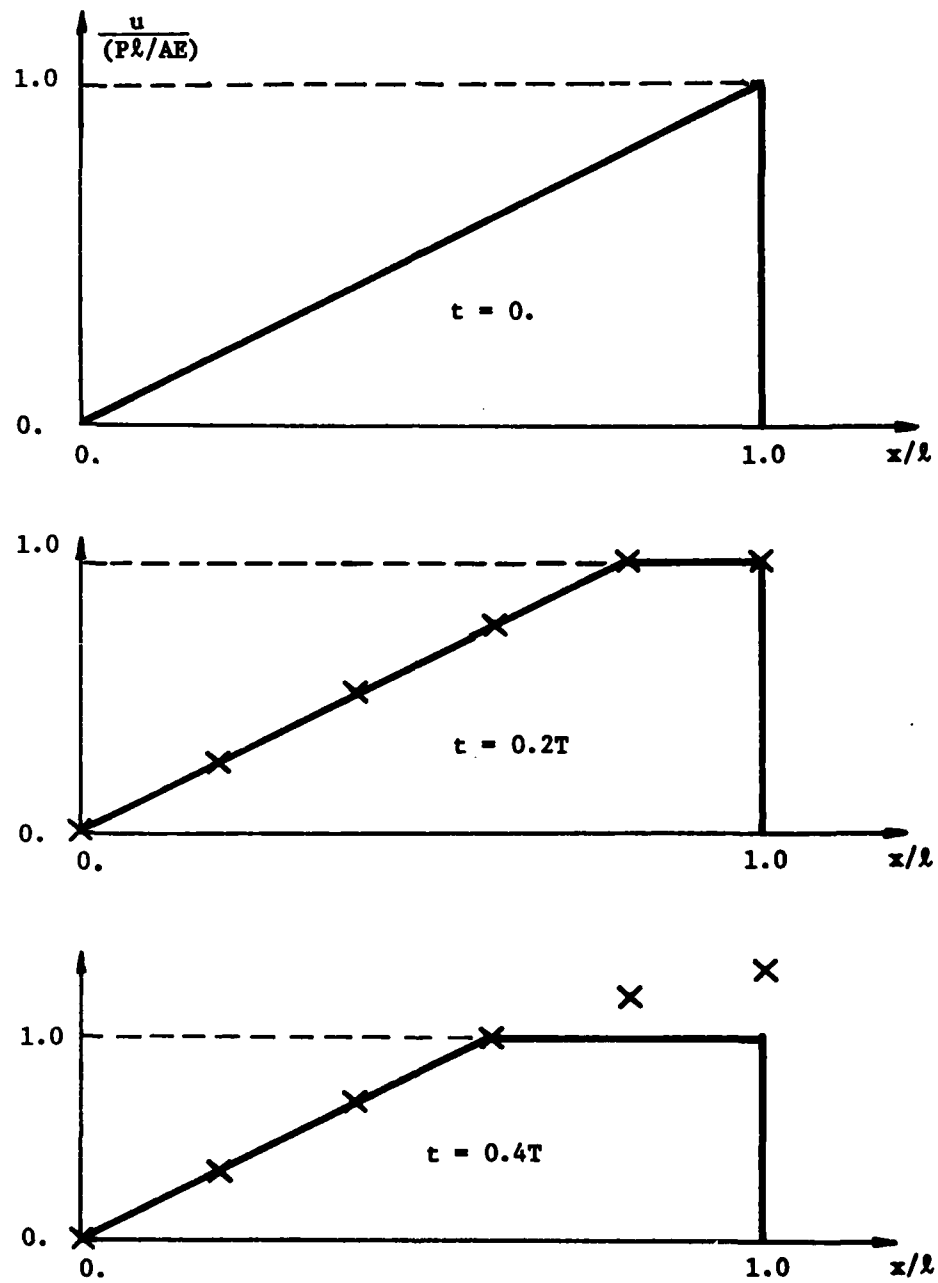


Figure 5. Displacement Solutions by the Unconstrained Variational Formulation ($t = 0, 0.2T$, and $0.4T$), and Comparison with Exact Solutions. Grid: 5×1 .

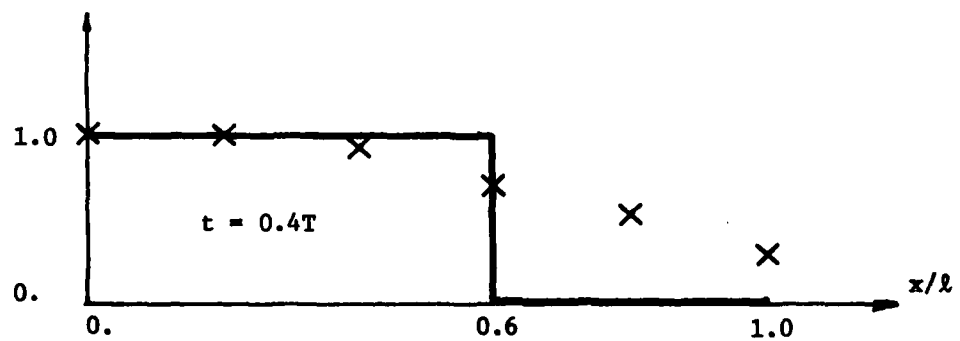
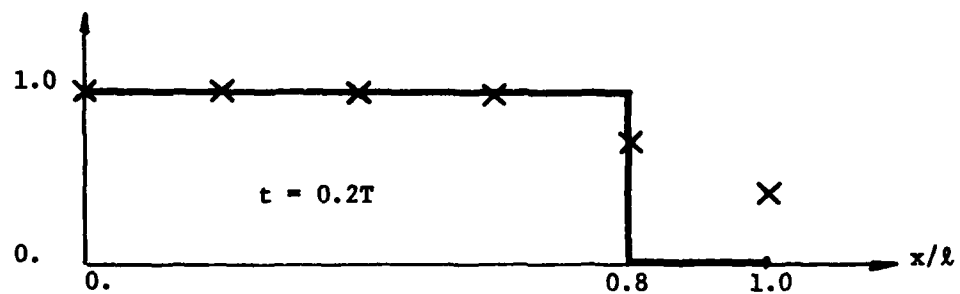
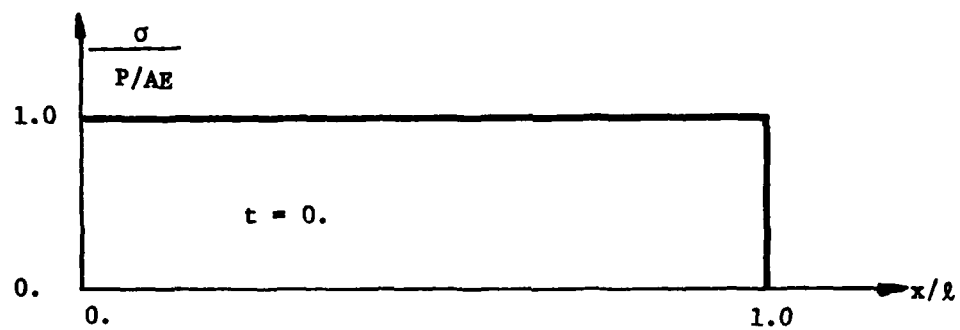


Figure 6. Stress Solutions by the Unconstrained Variational Formulation ($t = 0, 0.2T$, and $0.4T$) and Comparison with Exact Solutions. Grid: 5×1 .

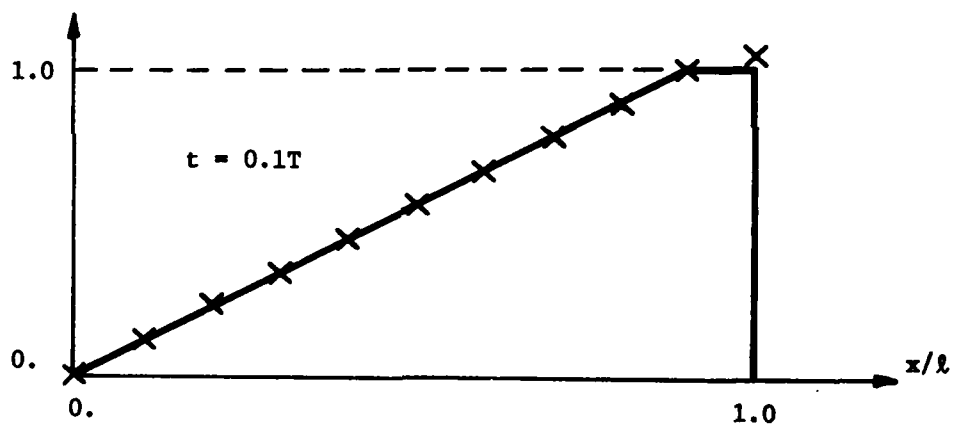
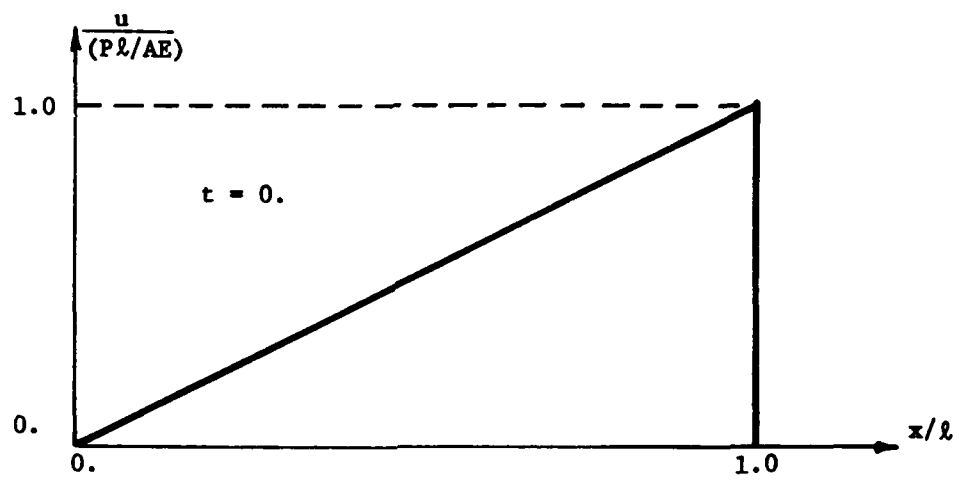


Figure 7. Displacement Solutions by the Unconstrained Variational Formulations ($t = 0.1T$) and Comparison with Exact Solutions. Grid: 10x1.

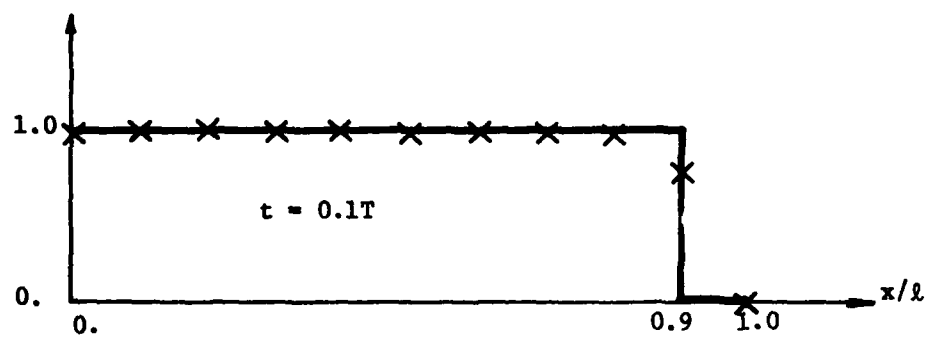
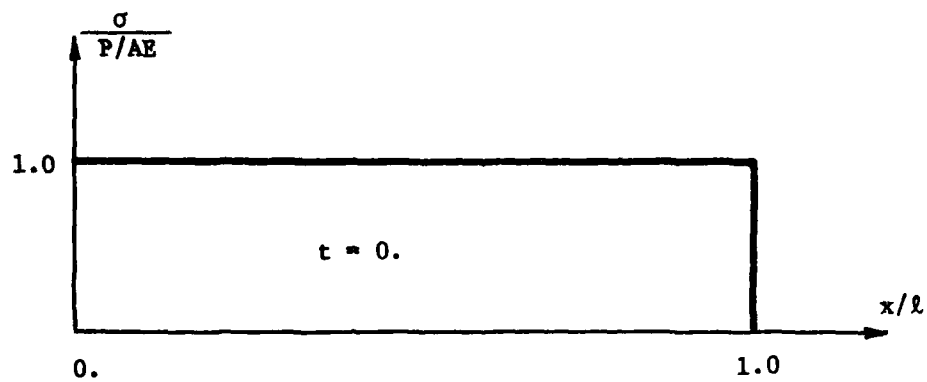


Figure 8. Stress Solution by the Unconstrained Variational Formulations ($t = 0.1T$) and Comparison with Exact Solutions. Grid: 10×1 .

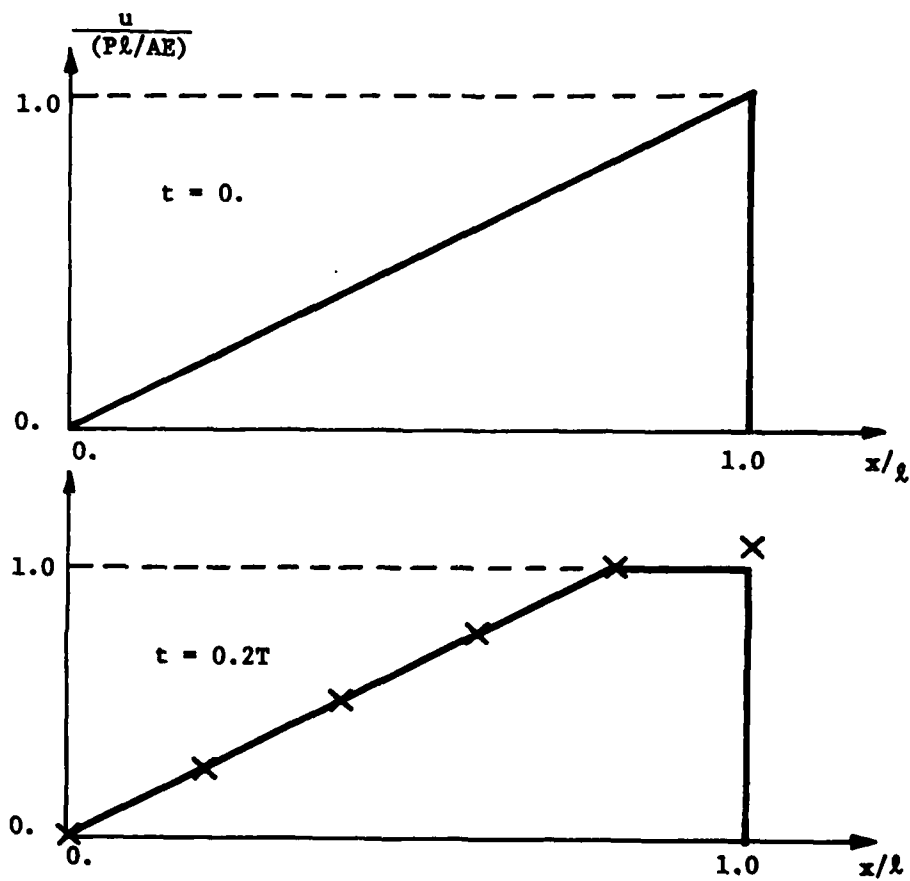


Figure 9. Displacement Solutions by the Constrained Variational Formulations ($t = 0.2T$) and Comparison with Exact Solutions. Grid: 5x1.

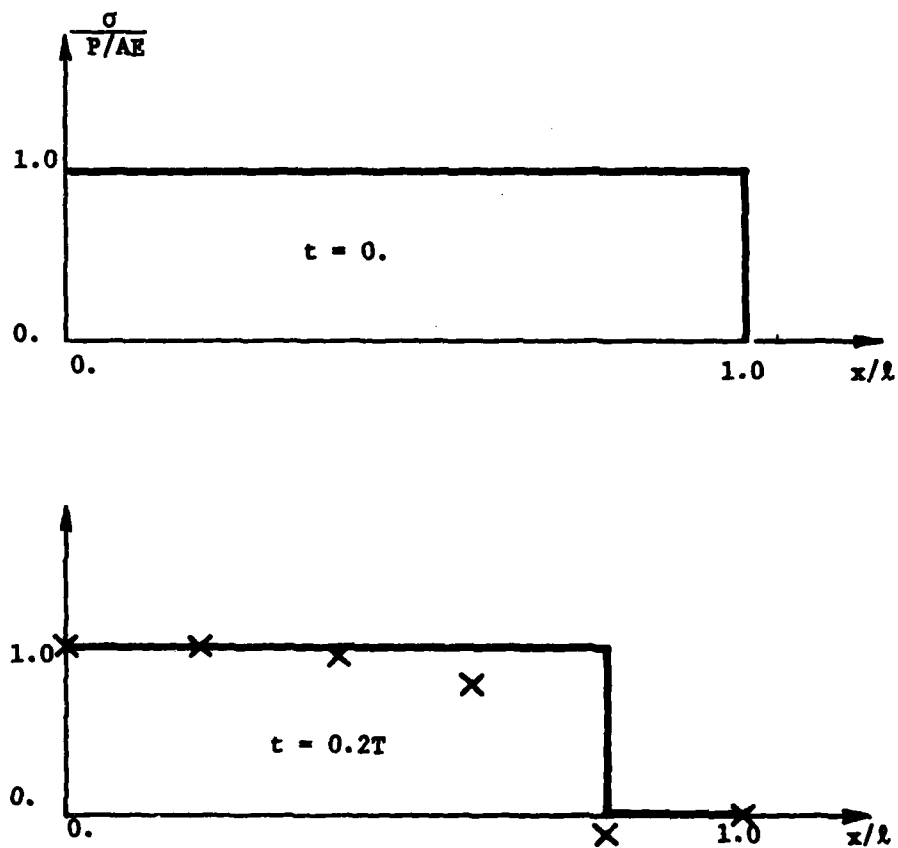


Figure 10. Stress Solution by the Constrained Variational Formulations ($t = 0.2T$) and Comparison with Exact Solutions. Grid: 5x1.

NUMERICAL SOLUTIONS USING ADJOINT VARIATIONAL
FORMULATION TO STRESS WAVE PROBLEMS

C. N. Shen and J. J. Wu
U.S. Army Armament Research and Development Command
Large Caliber Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189

ABSTRACT. A well known advantage of variational solution formulation to boundary value problems is that the differentiability requirements of the approximate solutions can be relaxed. For initial value problems, however, this advantage is somewhat diminished by the complication due to the appearance of the farther end condition. This complication can be eliminated by the use of an adjoint variational principle as we have demonstrated for a simple initial value problem in a previous paper. The more involved analysis for mixed initial boundary value problem has also been worked out.

The present paper deals with the numerical implementation of this more involved analysis in conjunction with cubic Hermite polynomials as the approximate functions. The specific example used for numerical results is the longitudinal stress wave of a uniform bar.

First, the adjoint principle associated with this problem is stated. It is followed by the discretized counterparts in spatial and temporal dimensions. The procedures involving the assemblage of the "mass" and "stiffness" matrices in the two dimensions are described. Due to the null variations of some adjoint variables, certain rows of the matrices are eliminated. Because certain variables are known at the boundaries, the unknown variables for the next interval of time can be computed by inversion of a band matrix in terms of their present values.

1. INTRODUCTION. The purpose of this paper is to employ the adjoint variational principle in the form of finite element formulation for solving the stress wave problems. The hyperbolic partial differential equation governs the motion is second order both in spatial and time domains.

$$Ly(x,t) + Q(x,t) = 0 \quad (1)$$

where

$$Ly = (\alpha y_t)_t + (\beta y_x)_x \quad (2)$$

We seek explicitly the numerical transient solutions of y , y_t , y_x and y_{xt} for assigned boundary and initial conditions. The term y_x will give the stress wave in a longitudinal bar. The study is the extension of previous work on initial and boundary value problems (ref. 1).

2. INTEGRAL OF BILINEAR EXPRESSION. The integral of a bilinear expression for a two dimensional problem having second order partial derivatives in both space and time can be written as

$$I = \int_{x_0}^{x_b} \int_{t_0}^{t_b} \Omega[y(x,t), \bar{y}(x,t)] dt dx \quad (3)$$

where $\Omega[y, \bar{y}]$ is a given bilinear expression in the form

$$\Omega[y, \bar{y}] = \alpha y_t \bar{y}_t + \lambda y_x \bar{y}_x \quad (4)$$

the quantity \bar{y} is the adjoint of y and the subscripts t and x indicate the partial derivatives of the functions y and \bar{y} .

Two different forms of integrals and end conditions can be obtained from Eq. (4). The first form is obtained by integrating by parts on the adjoint variable.

$$I = - \int_{x_0}^{x_b} \int_{t_0}^{t_b} \bar{y} Ly dt dx + \int_{x_0}^{x_b} \alpha \bar{y}_t y \Big|_{t_0}^{t_b} dx + \int_{t_0}^{t_b} \lambda \bar{y}_x y \Big|_{x_0}^{x_b} dt \quad (5)$$

where Ly is given in Eq. (2).

In addition, we can perform integration on the original variable to give

$$I = - \int_{x_0}^{x_b} \int_{t_0}^{t_b} y \bar{L} y dt dx + \int_{x_0}^{x_b} \alpha y_t \bar{y} \Big|_{t_0}^{t_b} dx + \int_{t_0}^{t_b} \lambda \bar{y}_x y \Big|_{x_0}^{x_b} dt \quad (6)$$

where

$$\bar{L} y = (\alpha \bar{y}_t)_t + (\lambda \bar{y}_x)_x \quad (7)$$

In a previous paper (ref. 1) we show that the bilinear concomitant D has to be identically zero, i.e.,

$$D = \int_{x_0}^{x_b} \int_{t_0}^{t_b} \bar{y} Ly dt dx - \int_{x_0}^{x_b} \int_{t_0}^{t_b} y \bar{L} y dt dx \quad (8)$$

By equating Eqs. (5) and (6) and solving for D in Eq. (8), we are converting the double integral into two single integrals in terms of the initial and boundary conditions.

We can express the quantity D as the sum of two parts for end conditions as D_1 and D_2 . Thus one defines

$$D = D_1 - D_2 \quad (9)$$

The terms in D_1 involve the initial conditions of y and \bar{y} as

$$\begin{aligned}
D_1 &= \int_{x_0}^{x_b} \left\{ \alpha_{ty} \bar{y} \Big|_{t_0}^{t_b} - \alpha_{ty} \bar{y} \Big|_{t_0}^{t_b} \right\} dx \\
&= \int_{x_0}^{x_b} \left\{ \alpha_b (y_{tb} \bar{y}_b - \bar{y}_{tb} y_b) - \alpha_0 (y_{t_0} \bar{y}_0 - \bar{y}_{t_0} y_0) \right\} dx
\end{aligned} \tag{10}$$

The terms in D_2 involve the boundary conditions of y and \bar{y} as

$$\begin{aligned}
D_2 &= \int_{t_0}^{t_b} \left\{ \lambda_{xy} \bar{y} \Big|_{x_0}^{x_b} - \lambda_{xy} \bar{y} \Big|_{x_0}^{x_b} \right\} dt \\
&= \int_{t_0}^{t_b} \left\{ \lambda_b (y_{xb} \bar{y}_b - \bar{y}_{xb} y_b) - \lambda_0 (y_{x_0} \bar{y}_0 - \bar{y}_{x_0} y_0) \right\} dt
\end{aligned} \tag{11}$$

In order that $D \equiv 0$ in Eq. (9) it is sufficient that

$$D_1 \equiv 0 \tag{12a}$$

and

$$D_2 \equiv 0 \tag{12b}$$

3. END CONDITIONS FOR THE ADJOINT SYSTEMS. In order to satisfy the two requirements in Eq. (12) we separate them in two parts. Let us consider first the time domain and assume that the adjoint variables are assigned as

$$\bar{y}_b = y_0, \quad \bar{y}_0 = y_b \tag{13}$$

$$\bar{y}_{tb} = -\alpha_b^{-1} \alpha_0 y_{t_0}, \quad \bar{y}_{t_0} = -\alpha_0^{-1} \alpha_b y_{tb} \tag{14}$$

$$\alpha_b \neq 0, \quad \alpha_0 \neq 0 \tag{15}$$

The above adjoint initial conditions satisfy the requirement that $D_1 \equiv 0$ in Eq. (10). Now we turn to the spatial domain and assume that that adjoint variables are

$$\bar{y}_b = y_b, \quad \bar{y}_0 = y_0 \tag{16}$$

$$\bar{y}_{xb} = y_{xb}, \quad \bar{y}_{x_0} = y_{x_0} \tag{17}$$

The above adjoint boundary conditions satisfy the requirement that $D_2 \equiv 0$ in Eq. (11).

By giving the appropriate values of these adjoint variables in terms of the original variables one may find that the requirement $D \equiv 0$ can be satisfied. This leads to the result (ref. 1) previously found as

$$J[\bar{y}, y] = \int_{t_0}^{t_b} \int_{x_0}^{x_b} Q y dt dx + \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{y} (Q + L y) dt dx = 0 \tag{18}$$

4. FIRST VARIATION. By taking variation on Eq. (18) we have

$$\delta J = \delta J[\bar{\delta y}] + \delta J[\delta y]$$

$$- \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{\delta y}(\bar{L}y) dt dx + \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{y}(L\delta y) dt dx = 0 \quad (19)$$

where

$$\delta J[\bar{\delta y}] = \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{\delta y}(\bar{L}y + Q) dt dx \quad (20)$$

and

$$\delta J[\delta y] = \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{\delta y}(\bar{L}y + Q) dt dx \quad (21)$$

Since $D \equiv 0$ in Eq. (8) the variation δD should be zero

$$\delta D = \delta D[\bar{\delta y}] + \delta D[\delta y] = 0 \quad (22)$$

Since the variation $\bar{\delta y}$ and δy are independent, then

$$\delta D[\delta y] = \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{y}(L\delta y) dt dx - \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{\delta y}(\bar{L}y) dt dx = 0 \quad (23)$$

which is the same as the last two terms in Eq. (19) which vanish. Thus

$$\delta J = \delta J[\bar{\delta y}] + \delta J[\delta y] = 0 \quad (24)$$

since the variation $\bar{\delta y}$ and δy are independent

$$\delta J[\bar{\delta y}] = \int_{t_0}^{t_b} \int_{x_0}^{x_b} \bar{\delta y}(\bar{L}y + Q) dt dx = 0 \quad (25)$$

where $L y$ is given in Eq. (2) and contains higher derivatives than the first partials in y . It is intended to include only lower order partial differentiation in y . This can be achieved by considering the variations of the bilinear expression I given by Eqs. (3) and (4) as

$$\delta I[\bar{\delta y}] = \int_{t_0}^{t_b} \int_{x_0}^{x_b} [\alpha y_t \bar{\delta y}_t + \lambda y_x \bar{\delta y}_x] dt dx \quad (26)$$

A different form of the above variation can be obtained from Eq. (5) as

$$\delta I[\bar{\delta y}] = - \int_{x_0}^{x_b} \int_{t_0}^{t_b} \bar{\delta y} L y dt dx + \int_{x_0}^{x_b} \bar{\delta y} \alpha y_t \Big|_{t_0}^{t_b} dx + \int_{t_0}^{t_b} \bar{\delta y} \lambda y_x \Big|_{x_0}^{x_b} dt \quad (27)$$

Equating Eqs. (26) and (27), solving for the term containing integral for $\delta \bar{y}_t$ and substituting into Eq. (25) we have

$$\begin{aligned} \delta J[\delta \bar{y}] = & \int_{x_0}^{x_b} \alpha y_t \delta \bar{y} \Big|_{t_0}^{t_b} dx + \int_{t_0}^{t_b} \lambda y_x \delta \bar{y} \Big|_{x_0}^{x_b} dt \\ & + \int_{x_0}^{x_b} \int_{t_0}^{t_b} \delta y Q dt dx - \int_{x_0}^{x_b} \int_{t_0}^{t_b} [\alpha y_t \delta \bar{y}_t + \lambda y_x \delta \bar{y}_x] dt dx = 0 \end{aligned} \quad (28)$$

This is the key equation which uses variational principle in solving a mixed initial and boundary value problem for a wave equation.

5. DISCUSSION OF THE VARIATIONAL EQUATION. Let us discuss the various terms in Eq. (28), the variational formulation of the wave equation, into three parts as follows.

(1) The initial conditions of the original variables are given and variations of the adjoints at the far end are zero. The first term in Eq. (28) contains the product of $y_t \delta \bar{y}$ evaluated at the initial condition $y_{t_0} \delta \bar{y}_0$ and at the final condition $y_{t_b} \delta \bar{y}_b$. Since the value of y_b are known by Eqs. (13) and (16). $\delta \bar{y}_b = 0$. That is, the variations of the adjoint variable at the far end are zero.

(2) The boundary conditions of the original variables and variation of the adjoints can be determined. The second term in Eq. (28) is the boundary term involving the variation $\delta \bar{y}$ and the variable y_x . For a longitudinal or a torsional bar the end conditions are

from Eq. (16) for the fixed end

$$y = 0 \quad \bar{y} = 0 \quad \delta \bar{y} = 0 \quad (29)$$

from Eq. (17) for the free end

$$y_x = 0 \quad \bar{y}_x = 0 \quad \delta \bar{y}_x = 0 \quad (30)$$

The variations in the adjoint variables shown in the last column coincide to the same end conditions in the original variables given in the first column, whether it is on the left or the right boundary.

(3) Interior Region - The last two terms in Eq. (28) give the interior region where the forcing function Q , the adjoint variation $\delta \bar{y}$, $\delta \bar{y}_t$, and $\delta \bar{y}_x$ and the variables y_t , and y_x are shown. No second order partial of y with respect to x is present. Thus the variables that are needed for the computation are y , y_t , and y_x . This requires a c^1 continuity in both spatial and time domain.

6. TRANSFORMATION OF COORDINATES. The integral signs in Eq. (28) can be converted into summation signs if discrete intervals for integration are used. We may take some scale factors to nondimensionalize the problem by giving

$$t_0 = 0, \quad t_b = 1 \quad 0 < t < 1 \quad (31)$$

$$x_0 = 0, \quad x_b = 1 \quad 0 < x < 1 \quad (32)$$

Moreover, Eq. (28) can be discretized by letting

$$\xi = Ht - i + 1 \quad 0 < \xi < 1 \quad i = 1, 2, \dots, H \quad (33)$$

$$\eta = Kx - j + 1 \quad 0 < \eta < 1 \quad j = 1, 2, \dots, K \quad (34)$$

where H and K are number of intervals for t and x respectively. Thus the partial derivatives are:

$$y_t = \frac{\partial y}{\partial t} = H \frac{\partial y}{\partial \xi} = Hy_\xi \quad (35)$$

$$y_x = \frac{\partial y}{\partial x} = \frac{\partial y}{\partial \eta} = Ky_\eta \quad (36)$$

Use of Eqs. (28), (31) through (36) then leads to

$$\begin{aligned} 0 &= \delta J[\bar{\delta y}] \\ &= \sum_{j=1}^K \frac{H}{K} \int_0^1 \alpha y_\xi(i, j) \delta y(i, j) d\eta \Big|_{t_0}^{t_b} \\ &\quad + \sum_{i=1}^H \frac{K}{H} \int_0^1 \lambda y_\eta(i, j) \delta y(i, j) d\xi \Big|_{x_0}^{x_b} \\ &\quad + \sum_{j=1}^K \sum_{i=1}^H \frac{1}{HK} \int_0^1 \int_0^1 Q \delta \bar{y}(i, j) d\xi d\eta \\ &\quad - \sum_{j=1}^K \sum_{i=1}^H \left\{ \frac{H}{K} \int_0^1 \int_0^1 \alpha y_\xi(i, j) \bar{\delta y}_\xi(i, j) d\xi d\eta + \frac{K}{H} \int_0^1 \int_0^1 \lambda y_\eta(i, j) \bar{\delta y}_\eta(i, j) d\xi d\eta \right\} \end{aligned} \quad (37)$$

7. SPLINE FUNCTION. We may express the variables $y(i, j)$ and $\bar{\delta y}(i, j)$ in Eq. (37) in terms of the (1×16) spline function $a^T(\xi, \eta)$ and the (16×1) node point function $y(i, j)$ as follows.

$$y(i, j)(\xi, \eta) = a^T(\xi, \eta) y(i, j) \quad (38)$$

where

$$a^T(\xi, \eta) = \{[a^1(\xi, \eta)]^T [a^2(\xi, \eta)]^T [a^3(\xi, \eta)]^T [a^4(\xi, \eta)]^T\} \quad (39)$$

and

$$\delta \bar{y}(i, j)(\xi, \eta) = a^T(\xi, \eta) \delta \bar{Y}(i, j) \quad (40)$$

A typical term for a product can be written as

$$\delta \bar{y}(i, j) y(i, j) = [\delta \bar{Y}(i, j)]^T a(\xi, \eta) a^T(\xi, \eta) Y(i, j) \quad (41)$$

Thus Eq. (37) becomes

$$\begin{aligned} \delta J(\delta y) = & \sum_{j=1}^K [\delta \bar{Y}(t_b, j)]^T P_{0\xi}(t_b) Y(t_b, j) \\ & - \sum_{j=1}^K [\delta \bar{Y}(t_0, j)]^T P_{0\xi}(t_0) Y(t_0, j) \\ & + \sum_{i=1}^H [\delta \bar{Y}(i, x_b)]^T P_{0\eta}(x_b) Y(i, x_b) \\ & - \sum_{i=1}^H [\delta \bar{Y}(i, x_0)]^T P_{0\eta}(x_0) Y(i, x_0) \\ & + \sum_{j=1}^K \sum_{i=1}^H [\delta \bar{Y}(i, j)]^T q(i, j) \\ & - \sum_{j=1}^K \sum_{i=1}^H [\delta \bar{Y}(i, j)]^T P(i, j) Y(i, j) = 0 \end{aligned} \quad (42)$$

where the coefficient P contains integrals involving the spline functions $a(\xi, \eta)$ and its partial derivatives as given in a previous paper (ref. 1).

8. GRID SYSTEMS FOR FINITE ELEMENT. We take a finite element represented by the (16×1) vector $\bar{Y}(i, j)$ which has a grid of four (4×1) vectors $Y_1(i, j)$ through $Y_4(i, j)$, thus

$$\bar{Y}(i, j) = \{[Y_1(i, j)]^T [Y_2(i, j)]^T [Y_3(i, j)]^T [Y_4(i, j)]^T\} \quad (43)$$

Each of the (4×1) vector has four components, consisting of the function, its first partials in both directions, and its mixed partial, as shown in Figure 1.

These vectors are,

$$\begin{aligned}
 Y_1(i,j) &= \begin{bmatrix} y(\xi_1, \eta_j) \\ y_\xi(\xi_1, \eta_j) \\ y_\eta(\xi_1, \eta_j) \\ y_{\xi\eta}(\xi_1, \eta_j) \end{bmatrix} & Y_3(i,j) &= \begin{bmatrix} y(\xi_1, \eta_{j+1}) \\ y_\xi(\xi_1, \eta_{j+1}) \\ y_\eta(\xi_1, \eta_{j+1}) \\ y_{\xi\eta}(\xi_1, \eta_{j+1}) \end{bmatrix} \\
 Y_2(i,j) &= \begin{bmatrix} y(\xi_{i+1}, \eta_j) \\ y_\xi(\xi_{i+1}, \eta_j) \\ y_\eta(\xi_{i+1}, \eta_j) \\ y_{\xi\eta}(\xi_{i+1}, \eta_j) \end{bmatrix} & Y_4(i,j) &= \begin{bmatrix} y(\xi_{i+1}, \eta_{j+1}) \\ y_\xi(\xi_{i+1}, \eta_{j+1}) \\ y_\eta(\xi_{i+1}, \eta_{j+1}) \\ y_{\xi\eta}(\xi_{i+1}, \eta_{j+1}) \end{bmatrix} \quad (44)
 \end{aligned}$$

We use the vertical direction for the temporal domain. If we increase the row index from i to $i+1$, then the grid point shifts down by one step and the following holds

$$Y_1(i+1,j) = Y_2(i,j) \quad Y_3(i+1,j) = Y_4(i,j) \quad (45)$$

If we increase the column index from j to $j+1$ then the grid point shifts to the right by one step and one obtains

$$Y_1(i,j+1) = Y_3(i,j) \quad Y_2(i,j+1) = Y_4(i,j) \quad (46)$$

Figure 2 shows the relationship of the grid system by assembly of finite elements in the horizontal direction, which is in the spatial domain.

9. ASSEMBLY OF MATRICES. In order to solve Eq. (42) by finite element method, assembly of matrices from local form into global form is necessary. For instance, the last term of Eq. (42) is taken as $-\delta J_p(\delta y)$. Then

$$\delta J_p(\delta y) = \sum_{j=1}^K \sum_{i=1}^H [\delta \bar{y}(i,j)] T_p(i,j) Y(i,j) \quad (47)$$

Since we know that the interval in time can be made as small as possible, with $H = 1$, we have

$$\begin{aligned}
\delta J_p(\delta y) &= \sum_{j=1}^K [\delta \bar{y}(1,j)] T_p(1,j) y(1,j) \\
&= \sum_{j=1}^K \{ [\delta \bar{y}_1(1,j)] T_1 [\delta \bar{y}_2(1,j)] T_2 [\delta \bar{y}_3(1,j)] T_3 [\delta \bar{y}_4(1,j)] T_4 \} \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \begin{bmatrix} y_1(1,j) \\ y_2(1,j) \\ y_3(1,j) \\ y_4(1,j) \end{bmatrix}
\end{aligned} \tag{48}$$

It is noted from Figure 2 that the variables can be indexed as

$$y_3(1,j) = y_1(1,j+1) = y_{2j+1} \tag{49}$$

$$y_4(1,j) = y_2(1,j+1) = y_{2j+2} \tag{50}$$

$$j=0,1,\dots,k$$

For $j = 0$,

$$y_1(1,1) = y_1, \quad y_2(1,1) = y_2 \tag{51}$$

For $j = k = 5$

$$y_3(1,5) = y_{11}, \quad y_4(1,5) = y_{12} \tag{52}$$

Also from Figure 2, the adjoint variations are

$$\delta \bar{y}_3(1,j) = \delta \bar{y}_1(1,j+1) = \delta \bar{y}_{2j+1} \tag{53}$$

$$\delta \bar{y}_4(1,j) = \delta \bar{y}_2(1,j+1) = \delta \bar{y}_{2j+2} \tag{54}$$

For $j = 0$,

$$\delta \bar{y}_1(1,j) = \delta \bar{y}_1, \quad \delta \bar{y}_2(1,1) = \delta \bar{y}_2 \tag{55}$$

For $j = k = 5$,

$$\delta \bar{y}_3(1,5) = \delta \bar{y}_{11}, \quad \delta \bar{y}_4(1,5) = \delta \bar{y}_{12} \tag{56}$$

Now the local matrix in Eq. (48) can be assembled into a global band matrix shown in Figure 3. Those elements not explicitly written are zeroes in Figure 3.

Since the adjoint variable y_b at the far end is assigned in terms of the known initial value y_0 , the variation is

$$\delta y_b = 0 \tag{57}$$

From Figure 2 we have

$$\bar{\delta Y}_2 = \bar{\delta Y}_4 = \bar{\delta Y}_6 = \bar{\delta Y}_8 = \bar{\delta Y}_{10} = \bar{\delta Y}_{12} = \bar{\delta Y}_{\text{EVEN}} = 0 \quad (58)$$

This is equivalent to deleting the even rows of the matrix in Figure 3. The deletion is marked in Figure 4. The number of relationships is reduced to half of the original dimension.

The variables Y_{ODD} in Figure 2 are the initial values of the problem which are supposed to be given. Thus, $Y_1, Y_3, Y_5, Y_7, Y_9, Y_{11}$ are all knowns. The coefficients related to these knowns should eventually be shifted to the right side of the equation.

10. FURTHER DELETIONS AND KNOWNs. Suppose we have a bar with the fixed end at the left. Then from Eq. (29) one obtains

$$y_0 = 0 \quad (59)$$

and

$$\bar{\delta y}_0 = 0 \quad (60)$$

The above equations translate to be

$$y(2,1) = 0 \quad (\text{known}) \quad (61)$$

and

$$\bar{\delta y}(1,1) = 0 \quad (\text{deletion}) \quad (62)$$

On the other hand we have a free end at the right. Then Eq. (30) gives

$$y_{xb} = 0 \quad (63)$$

and

$$\bar{\delta y}_{xb} = 0 \quad (64)$$

The above equations yield

$$y_{\eta}(2,6) = 0 \quad (\text{known}) \quad (65)$$

and

$$\bar{\delta y}_{\eta}(1,6) = 0 \quad (\text{deletion}) \quad (66)$$

Figure 5 gives the variation of adjoint variables. It shows two extra zero variations at the first row, $\bar{\delta y}(1,1)$ at the left and $\bar{\delta y}_{\eta}(1,6)$ at the right. We have also all zero variations on the second row. Figure 6 shows the known and unknown variables. There are two extra known variables in the second row due to boundary conditions, $y(2,1)$ at the left and $y_{\eta}(2,6)$ at the right. The first row gives all known initial conditions.

11. CONCLUSIONS. Direct computation of stress, i.e., numerical solution for first spatial derivatives of the displacement can be obtained directly. This is important if the problem has noisy components in the solution of the displacement. Computation can be made successively, i.e., the final values of the solution at the first stage in time can be used as the initial values of the second period in time. The variations of the adjoint variables at the far end in time for an initial-boundary value problem are zeroes. Deletion of

many rows in the assembled matrix is possible. The assembled matrix for computing is reduced to less than half size in linear dimension, from $(2n \times 2n)$ to $(n \times n)$. Hence, a bigger number of intervals in the spatial dimension can be handled. The reduced matrix is a band matrix which makes the storage requirement for computation much easier.

REFERENCES

1. Shen, C. N., "Method of Solution for Variational Principle Using Bicubic Hermite Polynomial," presented at the 27th Conference of Army Mathematicians, West Point, NY, June 1981.

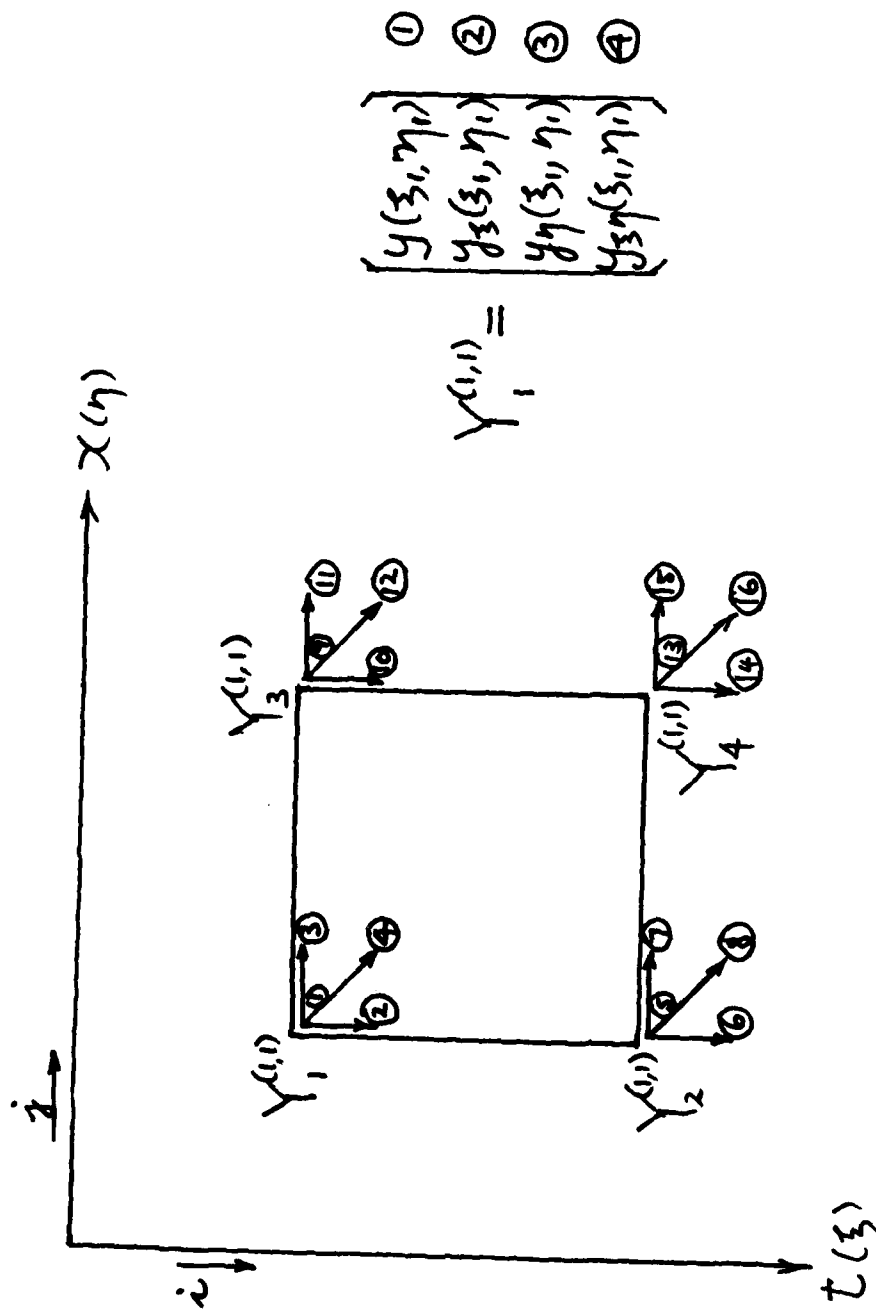


FIGURE 1 VECTORS IN A FINITE ELEMENT

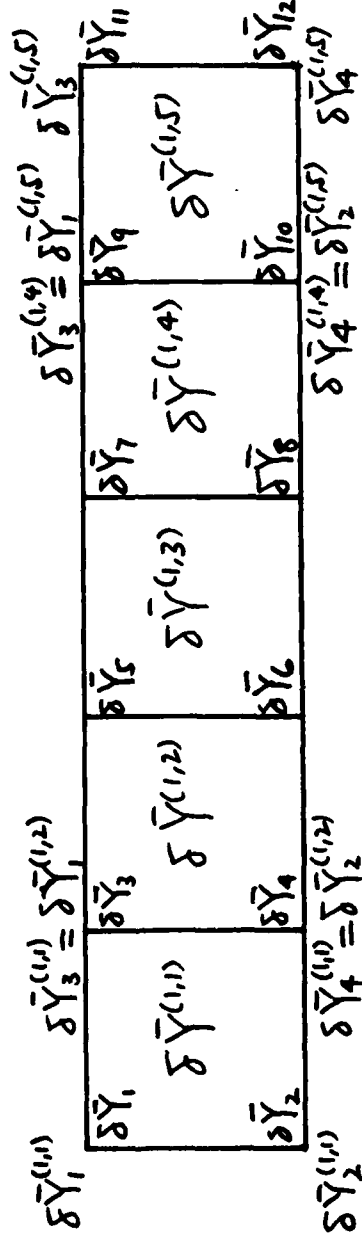
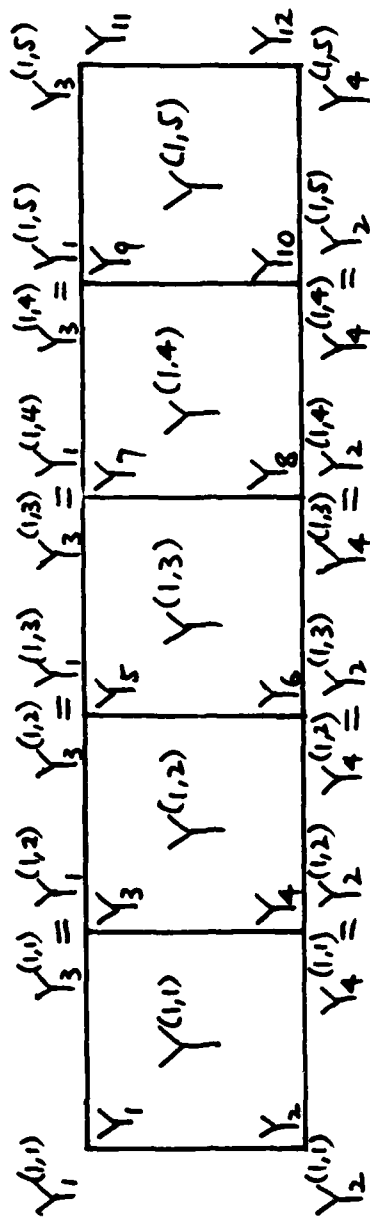


FIG 2 GRID SYSTEM BY ASSEMBLY
OF FINITE ELEMENTS

$$\begin{bmatrix} \delta \bar{Y}_1 \\ \delta \bar{Y}_2 \\ \delta \bar{Y}_3 \\ \delta \bar{Y}_4 \\ \delta \bar{Y}_5 \\ \delta \bar{Y}_6 \\ \delta \bar{Y}_7 \\ \delta \bar{Y}_8 \\ \delta \bar{Y}_9 \\ \delta \bar{Y}_{10} \\ \delta \bar{Y}_{11} \\ \delta \bar{Y}_{12} \end{bmatrix}^T \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \\ Y_8 \\ Y_9 \\ Y_{10} \\ Y_{11} \\ Y_{12} \end{bmatrix}$$

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \begin{bmatrix} P_{aa} & P_{ab} & P_{13} & P_{14} \\ P_{ba} & P_{bb} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \begin{bmatrix} P_{aa} & P_{ab} & P_{13} & P_{14} \\ P_{ba} & P_{bb} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} \begin{bmatrix} P_{aa} & P_{ab} & P_{13} & P_{14} \\ P_{ba} & P_{bb} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix}$$

$$\delta \bar{Y}_{EVEN} = 0 \quad P_{aa} = P_{11} + P_{33} \quad P_{ab} = P_{12} + P_{34} \quad Y_{ODD} \rightarrow \text{KNOWN}$$

$$P_{ba} = P_{21} + P_{43} \quad P_{bb} = P_{22} + P_{44}$$

FIGURE 3 MATRIX ASSEMBLY GLOBLE FORM

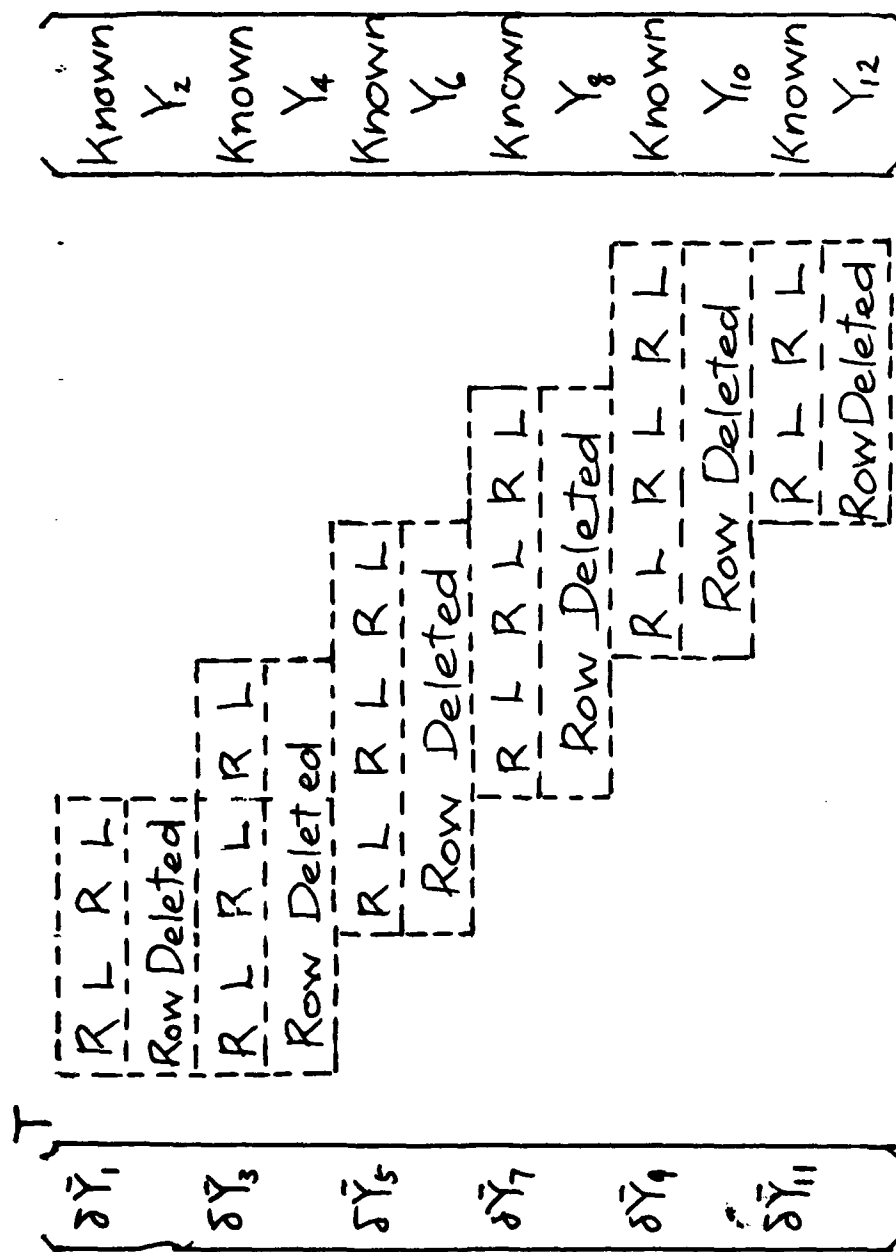
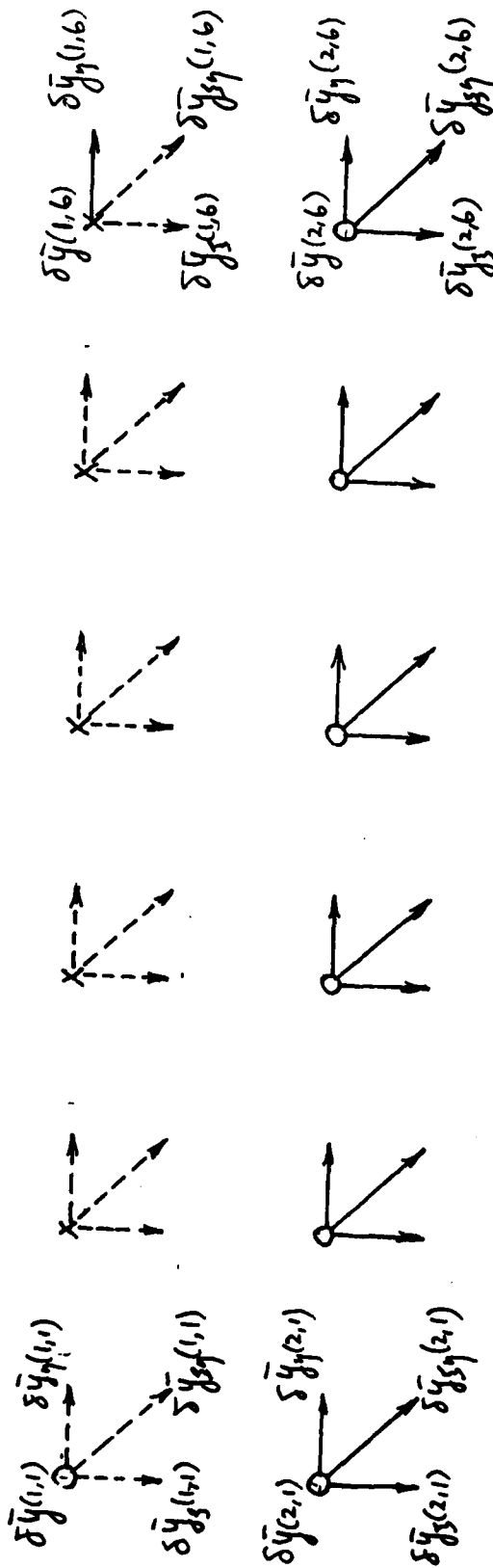


FIGURE 4 ROW DELETION & KNOWNS

FIXED END

FREE END

FIRST ROW: 2 NULL VARIATIONS \rightarrow 2 DELETED EQUATIONS
 22 ARBITRARY VARIATIONS \rightarrow 22 EQUATIONS



SECOND ROW: ALL FAR END VARIATIONS IN TIME ARE ZEROES
 24 DELETED EQUATIONS.

O ZERO \rightarrow ZERO X NOT ZERO \dashrightarrow NOT ZERO

FIGURES ZERO VARIATION OF ADJOINT VARIABLES

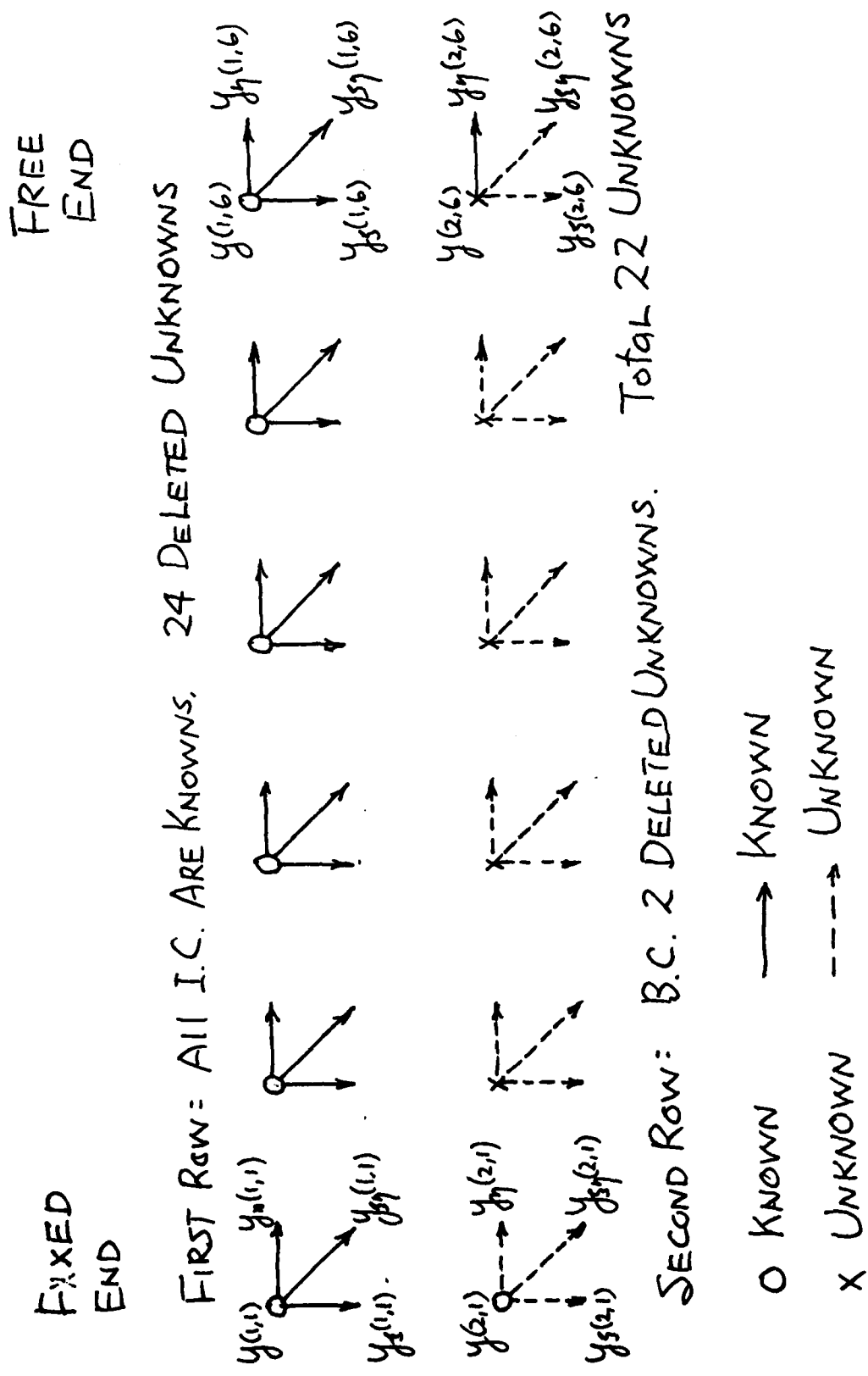


FIGURE 6 VARIABLES, KNOWN OR UNKNOWN

FINITE DIFFERENCE METHODS FOR THE STOKES AND NAVIER-STOKES EQUATIONS

John C. Strikwerda
Mathematics Research Center and
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706

ABSTRACT. This paper presents a new finite difference scheme for the Stokes equations and incompressible Navier-Stokes equations for low Reynold's number. The scheme uses the primitive variable formulation of the equations and is applicable with non-uniform grids and non-rectangular geometries. Several other methods of solving the Navier-Stokes equations are also examined in this paper and some of their strengths and weaknesses are described. Computational results using the new scheme are presented for the Stokes equations for a region with curved boundaries and for a disc with polar coordinates. The results show the method to be second-order accurate.

1. INTRODUCTION. In this paper we examine several common methods for solving the incompressible Navier-Stokes equations by finite differences and we present a new second-order accurate finite difference scheme for these equations. This new scheme is designed to be applied with non-uniform grids and non-orthogonal coordinate systems. Numerical experiments with the Stokes equations illustrate the versatility and accuracy of the scheme.

The steady-state Stokes equations on a domain Ω in \mathbb{R}^n are given by

$$\begin{aligned} (1.1) \quad & -\nabla^2 \vec{u} + \vec{\nabla} p = \vec{f}(x) \\ & \vec{\nabla} \cdot \vec{u} = g(x) \end{aligned}$$

and the steady-state Navier-Stokes equations are

$$\begin{aligned} (1.2) \quad & -R^{-1} \nabla^2 \vec{u} + (\vec{u} \cdot \vec{\nabla}) \vec{u} + \vec{\nabla} p = \vec{f}(x) \\ & \vec{\nabla} \cdot \vec{u} = g(x) \end{aligned}$$

where R is the Reynolds number. We will consider the systems (1.1) and (1.2) with Dirichlet boundary conditions

$$(1.3) \quad \vec{u}(x) = \vec{b}(x) \quad \text{on } \partial\Omega.$$

A necessary condition for (1.1) or (1.2) to have a solution is that the data $g(x)$ and $\vec{b}(x)$ satisfy the integrability condition

$$(1.4) \quad \int_{\Omega} g = \int_{\partial\Omega} \vec{b} \cdot \vec{n},$$

where \vec{n} is the outer unit normal to $\partial\Omega$. For the mathematical theory of the systems (1.1) and (1.2) we refer to Ladyzhenskaya (1963) and Teman (1979).

We will be concerned only with methods that solve the systems (1.1) and (1.2) in the primitive variables u and p and not with methods such as the vorticity and stream-function reformulation. Also our methods are applicable in two or three dimensions although our examples will be only in two dimensions.

We emphasize that the scheme presented here is designed to be easily applicable with non-rectangular geometries and non-uniform grids. The vast majority of papers on the numerical solution of the incompressible Navier-Stokes equations limit themselves to examples using rectangular geometry and uniform grids. By way of contrast, computations with the compressible Navier-Stokes equations routinely use non-rectangular geometries and non-uniform grids.

The outline of the remaining sections of the paper is as follows. In Section 2 we discuss the strengths and weaknesses of some common approaches to solving the systems (1.1) and (1.2) and in Section 3 we discuss finite difference schemes for these systems. The finite difference integrability condition is discussed in Section 4, and computational results are given in Section 5. The numerical examples of Section 5 demonstrate that the new scheme given here can be used to give second-order accurate solutions to the Stokes equations for non-rectangular geometries. To our knowledge no other finite difference schemes for the Stokes or incompressible Navier-Stokes equations in the primitive variables have been shown to be second-order accurate for non-rectangular geometries. Computations using the new scheme for the incompressible Navier-Stokes equations are currently being made and will be reported when complete.

2. SOLUTION TECHNIQUES. In this section we review some approaches to solving the Navier-Stokes and Stokes equations numerically. Few researchers have treated the system (1.2) in the given form, most have altered it in some way. Before examining the altered forms of (1.2) we look at the system in the given form.

The Stokes equations (1.1) and the Navier-Stokes equations (1.2) are elliptic systems of $n + 1$ equations in $n + 1$ dependent variables. The definition of an elliptic system, as given by Douglis and Nirenberg (1957), requires that the determinant of the principle symbol of the system not vanish for non-zero values of dual variables. For the Navier-Stokes equations the determinant of the principle symbol is

$$(2.1) \quad \det \begin{pmatrix} \frac{1}{R} |\xi|^2 T_n & i\xi \\ i\xi^T & 0 \end{pmatrix} = |\xi|^{2n}$$

which is non-zero for $|\xi| \neq 0$. Moreover, since the determinant is a polynomial of degree $2n$ in the variables $\xi = (\xi_1, \dots, \xi_n)$ the system requires n boundary conditions at each point of the boundary (Agmon, Douglis and Nirenberg (1964)). These boundary conditions will usually be Dirichlet or Neumann conditions on the velocity u .

One of the most common ways of modifying the Navier-Stokes equations (1.2) is to replace it by the system

$$(2.2) \quad \begin{aligned} -R^{-1} \nabla^2 \vec{u} + (\vec{u} \cdot \nabla) \vec{u} + \nabla p &= \vec{f}(x) \\ \nabla^2 p &= \nabla \cdot \vec{f} + R^{-1} \nabla^2 g - \sum_{i,j} u_j^i u_i^j - \vec{u} \cdot \nabla g \end{aligned}$$

The last equation of (2.2) is obtained by taking the divergence of the first equation of (1.2) and then using the last equation of (1.2) to eliminate the divergence of velocity. The system (2.2) has the advantage over (1.2) in that, when discretized, it can be solved using standard methods for inverting the discrete Laplacian. However, the system (2.2) has a grave disadvantage in that it requires $n + 1$ boundary conditions, one for each elliptic equation, as opposed to (1.2) which requires only n boundary conditions. Thus any attempt to solve (1.2) via (2.2) would require some means of determining the correct additional boundary condition. Without the correct condition solutions of (2.2) will not be solutions of (1.2).

Roache (1972, p. 194) suggests that the additional boundary condition be given by the normal derivative of pressure as determined by the first equation of (1.2) or (2.2) evaluated on the boundary. This, however, is not satisfactory as a boundary condition since it is not independent of the system of differential equations. Roache's suggestion leaves the system (2.2) underdetermined.

Another boundary condition which is commonly used along boundaries corresponding to physical surfaces is to set the normal derivative of the pressure to zero, which is valid in the limit for high Reynolds number flow. With this boundary condition and (1.3) the system (2.2) has the proper number of boundary conditions, however, its solutions are not solutions of (1.2).

As one would expect, the methods using (2.2) or similar systems have difficulty with the accuracy of the pressure field and with satisfying the incompressibility condition on the velocities (see for example the work by Boney, Hefner, Hirsh, and Zoby reported in Rubin and Harris (1975)).

The above mentioned difficulties are seen in computations with the time-dependent Navier-Stokes equations as well. Roache (1972) has a discussion of the difficulties of obtaining a zero divergence for the velocity field when using the above approach for time dependent flows (see also Harlow and Welch (1964)).

Because of these difficulties, it seems best not to use the derived system (2.2) but to use the original system (1.2).

Another approach to solving the Navier-Stokes equations (1.2) is the artificial compressibility method. The basic idea of this method is to solve a time-dependent system of equations, whose steady-state solutions solve (1.2), until a steady state is reached. Methods have been proposed by Chorin (1967) and Yanenko (1967). The convergence rate of these methods is dependent on the choice of finite difference method used to solve the system. Moreover, as will be discussed in Section 4, it may happen that the finite difference

equations do not have a steady-state solution, so the method cannot converge. Taylor and Ndefo (1970) reported difficulty in getting Yanenko's method to converge, most likely because there was no solution.

Another common method is to use the "parabolized" Navier-Stokes equations in which the second-derivatives in the stream-wise direction are removed. Because of its limited applicability and uncertain justification we will not discuss this method here except to note that often an analogue of (2.2) is derived and thus some of our observations on (2.2) also apply to the parabolized equations. Raithby and Schneider (1979) discuss these difficulties for three-dimensional flow problems.

3. FINITE DIFFERENCE SCHEMES. In this section we discuss the staggered mesh and central finite difference schemes for (1.1) and (1.2) and introduce a new scheme. The second-order accurate staggered mesh scheme for a uniform cartesian grid assigns the values of each of the velocity components and the pressure to different interlaced grids. In two dimensions with velocity components u and v , one may assign values of u to grid locations

$((i + \frac{1}{2})h, jh)$, values of v to $(ih, (j + \frac{1}{2})h)$, and values of p to (ih, jh) , e.g. Harlow and Welch (1965), Patankar and Spalding (1972), Raithby and Schneider (1979), Brandt and Dinar (1979). This method works very well as long as the geometry is rectangular and the grid is uniform. Non-uniform grids and grid mapping techniques cannot be conveniently handled.

The staggered mesh schemes also have some difficulty at boundaries. For example, when both velocity components are specified at a boundary then that velocity component whose mesh lines do not lie on the boundary requires some special treatment.

The central difference scheme on a uniform rectangular mesh assigns values of all the variables to each grid point. The divergence and gradient operators are approximated using central differences and the Laplacian is approximated by the standard five-point discrete Laplacian. Central difference schemes have been used by Chorin (1967, 1968) in time-dependent calculations.

An important concept for finite difference schemes for elliptic systems such as (1.1) and (1.2) is that of regularity (see Bube and Strikwerda (1980), and also Frank (1968), Brandt and Dinar (1979)). Regular schemes give rise to regularity estimates analogous to those in the theory of elliptic systems of differential equations. Solutions to regular difference schemes will in general be smoother than solutions to non-regular schemes and also will be more accurate approximations to the solutions of the differential equations.

The central difference scheme is non-regular (Bube and Strikwerda (1980)), which results in non-smooth solutions. The lack of smoothness is most noticeable in the pressure. The staggered mesh scheme is regular. The advantage of the central difference scheme is that it is easily implemented with non-uniform grids as introduced by coordinate changes.

It should be emphasized that none of the difficulties mentioned above are insurmountable. Both the staggered mesh and central differencing schemes have been used and often quite successfully. However we will consider a new scheme

which incorporates both regularity and ease of implementation with coordinate grid mapping techniques.

Before introducing the new scheme we will discuss the concept of regularity for difference schemes as given in Bube and Strikwerda (1980). A difference operator A may be written as

$$Af(x) = \sum_{\mu} a_{\mu}(h, x) T^{\mu} f(x) ,$$

where T^{μ} is the translation operator given by

$$T^{\mu} f(x_v) = f(x_{v+\mu})$$

for multi-indices v and μ .

The symbol of A is given by

$$a(h, x, \zeta) = \sum_{\mu} a_{\mu}(h, x) e^{i\mu \cdot \zeta} .$$

For example, the first-order central difference operator in the k -th coordinate direction has symbol

$$\frac{e^{i\zeta_k} - e^{-i\zeta_k}}{2h_k} = ih_k^{-1} \sin \zeta_k$$

and the standard second-order accurate Laplacian in n -variables has the symbol

$$- \sum_{k=1}^n 4h_k^{-2} \sin^2 \frac{1}{2} \zeta_k .$$

A finite difference scheme for the Stokes equations is regular elliptic if the determinant of the matrix of symbols of the scheme vanishes only for ζ equal to zero modulo 2π . For the Stokes equations with central differencing, and $\Delta x = \Delta y = h$, this determinant is

$$\det \begin{pmatrix} 4h^{-2}(\sin^2 \frac{1}{2} \zeta_1 + \sin^2 \frac{1}{2} \zeta_2) & 0 & ih^{-1} \sin \zeta_1 \\ 0 & 4h^{-2}(\sin^2 \frac{1}{2} \zeta_1 + \sin^2 \frac{1}{2} \zeta_2) & ih^{-1} \sin \zeta_2 \\ ih^{-1} \sin \zeta_1 & ih^{-1} \sin \zeta_2 & 0 \end{pmatrix}$$

$$= 4h^{-4}(\sin^2 \frac{1}{2} \zeta_1 + \sin^2 \frac{1}{2} \zeta_2)(\sin^2 \zeta_1 + \sin^2 \zeta_2) .$$

This determinant vanishes for the dual variables ζ_1 and ζ_2 equal to π , and thus the scheme is not regular. One sees that the non-regularity comes from the form of differencing used for the gradient and divergence terms. Our new scheme is a modification of the central differencing scheme so as to make the scheme regular.

The new scheme we consider will be called the regularized central difference scheme. In this scheme the derivatives of pressure are approximated as

$$(3.2) \quad \frac{\partial p}{\partial x_k} \approx \delta_{k0} p - \alpha h_k^2 \delta_{k-} \delta_{k+}^2 p$$

and the first derivatives of the velocity in the divergence equation are approximated as

$$(3.3) \quad \frac{\partial u^k}{\partial x_k} \approx \delta_{k0} u^k - \alpha h_k^2 \delta_{k+} \delta_{k-}^2 u^k,$$

where α is a non-zero constant and δ_{k0} , δ_{k+} and δ_{k-} are the centered, forward, and backward divided differences, respectively. The Laplacian is approximated with the usual five-point scheme. For a cartesian grid in two dimensions the determinant of the symbol is

$$\det \begin{pmatrix} 4h^{-2}(\sin^2 \frac{1}{2} \zeta_1 + \sin^2 \frac{1}{2} \zeta_2) & 0 & d(\zeta_1) \\ 0 & 4h^{-2}(\sin^2 \frac{1}{2} \zeta_1 + \sin^2 \frac{1}{2} \zeta_2) & d(\zeta_2) \\ -\overline{d(\zeta_1)} & -\overline{d(\zeta_2)} & 0 \end{pmatrix}$$

$$= 4h^{-2}(\sin^2 \frac{1}{2} \zeta_1 + \sin^2 \frac{1}{2} \zeta_2)(|d(\zeta_1)|^2 + |d(\zeta_2)|^2)$$

where

$$d(\zeta) = ih^{-1} \sin \zeta - \alpha h^{-1/2} i \zeta (2i \sin \frac{1}{2} \zeta)^3$$

$$= 2ih^{-1} \sin \frac{1}{2} \zeta (\cos \frac{1}{2} \zeta + 4\alpha e^{1/2} i \zeta \sin^2 \frac{1}{2} \zeta).$$

Since $d(\zeta)$ is not zero for any value of ζ , when α is non-zero, the scheme is regular. Note that for α equal to one-sixth the approximations (3.2) and (3.3) are third-order accurate.

Since the regularized central difference scheme is a variant of the central difference scheme it is easy to implement with coordinate maps. At those boundary points where the correction term would require points beyond the boundary we use the correction term which interchanges the forward and backward operators. This scheme also requires the use of extrapolation to

compute the pressure values on the boundary. It has been found that third order extrapolation gave quite good results, e.g.

$$(3.4) \quad p_{0j} = 3p_{1j} - 3p_{2j} + p_{3j}$$

at the boundary $x = 0$ in two dimensions.

A number of first-order accurate schemes for the Stokes and Navier-Stokes equations have been presented e.g. Kzivickii and Ladyzhenskaya (1966) and Temam (1979, p. 48). In this paper we are concerned only with second-order accurate schemes.

4. THE INTEGRABILITY CONDITION. Each of the schemes for the Stokes equations which have been discussed in the previous section can be written as

$$(4.1) \quad \begin{aligned} a) \quad L_h \vec{u}_h + G_h p_h &= \vec{f}_h \\ b) \quad D_h \cdot \vec{u}_h &= g_h \end{aligned} \quad \text{on } \Omega_h$$

with Dirichlet boundary conditions

$$(4.1) \quad c) \quad \vec{u}_h = \vec{b}_h \quad \text{on } \partial\Omega_h.$$

The difference operators L_h , G_h , and D_h are approximations to the differential operators in (1.1). The discrete functions \vec{f}_h , g_h , and \vec{b}_h are approximations to f , g and b on the mesh Ω_h , where h is some measure of the fineness of the mesh Ω_h .

Now let us compare the system (4.1) with the system (1.1). First note that if G_h is a consistent approximation to the gradient then the discrete pressure p_h is determined only up to a constant. This means that the system of linear equations (4.1) does not have full column rank. If there are as many equations in (4.1) as there are unknowns, and this is the case for each scheme we've considered, then the system (4.1) does not have full row rank either. This implies that there is a constraint which the data must satisfy to guarantee a solution, in particular, the discrete integrability condition analogous to (1.4) must be satisfied.

There are at least two ways to satisfy the discrete integrability condition. The first method would be to analyze the matrix corresponding to (4.1) and determine the null space of the adjoint matrix. If the data is constrained to be orthogonal to this null space then a solution will exist. This approach is impractical for many situations especially if coordinate changes have been employed since then the matrices are not easy to analyze.

A second approach, which will be adopted here, is to replace (4.1b) by

$$(4.1b') \quad D_h \cdot \vec{u}_h = g_h + \delta_h$$

where δ_h is a constant chosen to guarantee a solution. The value of δ_h

must be determined as part of the solution. As shown in the examples in Section 5 δ_h is at least $O(h^2)$ for the regularized central scheme. We will refer to the equations (4.1a, b', c) as (4.1').

It is interesting to note that for the staggered mesh scheme on a uniform grid one can easily satisfy the discrete integrability condition since the calculus of finite differences mimics the differential calculus very closely, see e.g. Kzivickii and Ladyzhenskaya (1966). Also, Ghia, Hankey, and Hodge (1977) mention being unable to obtain a solution to the discrete Navier-Stokes equations for certain situations. We conjecture that this difficulty was caused by the discrete integrability condition not being satisfied.

There is the possibility that the null space of the discrete operator of (4.1) has dimension greater than one. The regularized central scheme with the third-order extrapolation (3.4) appears to have only a one-dimensional null space. However, for α equal to zero numerical experiments indicate that there are solutions which are effectively null vectors in that they solve (4.1) with f_h and g_h smaller than the norm of the solution by a factor proportional to h or h^2 . The dimension of the space of nearly null vectors and null vectors appears to be four for the central differencing scheme. These vectors correspond to the four zeroes of the determinant of the symbol of the difference operator.

These nearly null vectors and null vectors, other than the usual constant pressure null solution, make solving the discrete system very difficult. On the other hand the regular discrete systems can be solved easily by the iterative procedure given in Strikwerda (1982).

5. COMPUTATIONAL RESULTS. In this section we present the results of testing the new scheme described in Section 3. In the examples discussed here the discrete Stokes equations were solved using test problems which illustrate various features of the schemes. For each example an exact analytical solution is known and the approximate solutions were compared to the exact solutions to study the accuracy of the method. The value of α , the regularity parameter, was one-sixth in all cases.

For the first test problem the Stokes equations were solved on the unit square with a uniform grid. The exact solution is

$$\begin{aligned} u &= (2\pi)^{-1} \sin \pi x \cos \pi y \\ v &= (2\pi)^{-1} \cos \pi x \sin \pi y \\ p &= \cos \pi x \cos \pi y \end{aligned} \tag{5.1}$$

with $\vec{f} = 0$ and $g = \cos \pi x \cos \pi y$. For this example both the accuracy and symmetry of the solution were checked. The symmetry was checked to study the effect of the nonsymmetric regularizing term on the symmetry of the solution. The symmetry was measured by computing the quantities $\text{sym}(u)$ and $\text{sym}(p)$ given by

$$\text{sym}(u) = \left(\sum_{i,j=0}^N (u_{ij} + u_{N-i,N-j})^2 \right)^{1/2} / \|u\|_2 \quad (5.2)$$

$$\text{sym}(p) = \left(\sum_{i,j=0}^N (p_{ij} - \bar{p})^2 \right)^{1/2} / \|p - \bar{p}\|_2$$

for an $(N+1) \times (N+1)$ grid. The quantity \bar{p} is the average value of the p_{ij} and the norm is the ℓ^2 -norm, e.g.

$$\|u\|_2 = \left(\sum_{i,j} u_{ij}^2 \right)^{1/2}.$$

The second test problem demonstrates the ability of the scheme to produce second-order accurate solutions on a non-rectangular region. The exact solution is

$$u = \xi^2 + \eta^2$$

$$v = -2\xi\eta + \eta^2$$

$$p = 4\xi + 2\eta$$

on the region Ω which is the image of the unit square under the mapping

$$\xi = x \cosh(y)$$

$$\eta = y - x^2$$

for (x,y) in the unit square, i.e. $0 \leq x, y \leq 1$. Thus the equations being solved on the unit square were

$$\begin{aligned} & x_\xi(x_\xi u_x)_x + x_\xi(y_\xi u_y)_x + y_\xi(x_\xi u_x)_y + y_\xi(y_\xi u_y)_y \\ & + x_\eta(x_\eta u_x)_x + x_\eta(y_\eta u_y)_x + y_\eta(x_\eta u_x)_y + y_\eta(y_\eta u_y)_y - x_\xi p_x - y_\xi p_y = 0 \end{aligned}$$

for the first equation, with the second being similar, and

$$x_\xi u_x + y_\xi u_y + x_\eta v_x + y_\eta v_y = 0$$

for the third equation. The regularizing terms were added only to the terms corresponding to p_x in the first equation, p_y in the second, and u_x and v_y in the third.

In the third test problem the Stokes equations were solved on a disc using polar coordinates with uneven grid spacing in both the radial and angular direction. The exact solution is

$$u = r^3 \sin 2\theta$$

$$v = 2r^3 \cos 2\theta$$

$$p = 6r^2 \sin 2\theta$$

with f and g being zero. The uneven grid was given by

$$r_i = .75 \rho_i + .25 \rho_i^2$$

$$\theta_j = \varphi_j - .25 \sin \varphi_j$$

where ρ_i and φ_j were evenly spaced in the interval $[0,1]$ and $[0,2\pi]$ respectively. This uneven spacing was chosen merely to show the versatility of the scheme and is not intended to give a better resolution of the solution.

For completeness we give the Stokes equations in polar coordinates

$$\begin{aligned} r^{-1}(ru_r)_r + r^{-2}u_{\theta\theta} - r^{-2}u - 2r^{-2}v_{\theta} - p_r &= 0 \\ (5.5) \quad r^{-1}(rv_r)_r + r^{-2}v_{\theta\theta} - r^{-2}v + 2r^{-2}u_{\theta} - r^{-1}p_{\theta} &= 0 \\ r^{-1}(ru)_r + r^{-1}v_{\theta} &= 0. \end{aligned}$$

The difference formulas used in the numerical experiments were all second order accurate. As an example of the formulas, the term $r^{-1}(ru)_r$ was differenced as

$$\left(\frac{r_{i+1} + r_i}{r_{i+1} - r_i}\right)(u_{i+1,j} - u_{i,j}) - \left(\frac{r_i + r_{i-1}}{r_i - r_{i-1}}\right)(u_{i,j} - u_{i-1,j}) / \frac{1}{2}(r_{i+1}^2 - r_{i-1}^2).$$

The results of the numerical experiments are shown in the following tables. Each table lists the errors incurred for grids with $N + 1$ points on a side for values of N of 20, 30, 40 and 60. Tables I, II and III list the relative errors for test problems 1, 2 and 3, respectively, and Table I also shows the symmetry errors for problem 1. The relative errors are measured in the l^2 -norm i.e.

$$\text{err}(u) = \left(\sum (u_{ij} - u(x_i, y_i))^2 \right)^{1/2} / \|u\|_2.$$

Also shown is the value of δ_h which is described in Section 4. Table IV displays the behavior of the δ_h error as the grid resolution is increased. The numbers shown are values of

$$-\frac{\log(\text{err}_1/\text{err}_2)}{\log(N_1/N_2)}$$

where err_1 and err_2 are the errors for grids of $N_1 + 1$ and $N_2 + 1$ points on a side, respectively. This value should be approximately 2.0 for a second-order scheme. The error reductions are shown for u , p and δ_h . The other velocity component had a similar error behavior in all the examples. All of the solutions were computed by the iterative method given in Strikwerda (1982).

That some of the errors were better than second-order accurate for test problems 1 and 3 can be attributed to the third-order accurate difference formulas used for the gradient and divergence terms. One might expect that some of the errors would behave as third-order errors for some value of N_1 and N_2 . However, since the discrete Laplacian is second-order accurate, for N large enough the total scheme should be second-order accurate. It is not clear why δ_h should behave as a fourth-order error as seen in test problem 3 and for some values of N_1 and N_2 in test problem 1. Test problem 2 was no better than second-order accurate since the gradient and divergence were only second-order accurate. The third-order differences were only used on those terms which were necessary to achieve regularity of the scheme. The results show conclusively that the scheme has overall second-order accuracy.

6. CONCLUSION. In this paper we have examined several finite difference methods for the steady Stokes and incompressible Navier-Stokes equations in primitive variables. We have shown that the regularized centered difference scheme is second-order accurate and useful with non-rectangular regions. Although the numerical experiments were done using the Stokes equations, for which exact solutions were available, we believe the regularized central scheme is equally useful with the incompressible Navier-Stokes equations at moderate Reynolds number.

REFERENCES

- S. Agmon, A. Douglis, L. Nirenberg (1964). Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions, II, Comm. Pure Appl. Math. 17, pp. 35-92.
- A. Brandt, N. Dinar (1979). Multi-grid solutions to elliptic flow problems, Proc. of Conference on Numerical Solutions of Part. Diff. Equations, Math. Res. Center, Madison, WI, October 1978.
- K. Bube, J. Strikwerda (1980). Interior regularity estimates for elliptic systems of difference equations, ICASE Report 8-28, SIAM J. Num. Anal. to appear.
- A. Chorin (1967). A numerical method for solving incompressible viscous flow problems, J. Comp. Phys., 2, pp. 12-26.
- A. Chorin (1968). Numerical solution of the Navier-Stokes equations, Math. Comput., 22, pp. 745-762.
- A. Douglis, L. Nirenberg (1955). Interior estimates for elliptic systems of partial differential equations, Comm. Pure Appl. Math., 8, pp. 503-538.

- L. Frank (1968). Difference operators in convolutions, Soviet Math. Dokl., 9, pp. 831-834.
- K. N. Ghia, W. L. Hankey, J. K. Hodge (1977). Study of incompressible Navier-Stokes equations in primitive variables using implicit numerical technique, AIAA paper 77-648.
- F. H. Harlow and J. E. Welch (1964). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics of Fluids, 8, pp. 2181-2189.
- A. Kzivickii, O. A. Ladyzhenskaya (1966). The method of nets for the non-stationary Navier-Stokes equations, Proc. of the Steklov Inst., 92, pp. 105-112.
- O. A. Ladyzhenskaya (1963). The Mathematical Theory of Viscous Incompressible Flows, Translated by R. A. Silverman, Gordon and Breach, New York.
- S. V. Patankar, D. B. Spalding (1972). A calculation procedure for heat, mass, and momentum transfer in three dimensional parabolic flows, Internat. J. Heat and Mass Trans., 15, pp. 1787-1806.
- G. D. Raithby, G. E. Schneider (1979). Numerical solution of problems in incompressible fluid flow: Treatment of the velocity-pressure coupling, Num. Heat Transf., 2, 417-440.
- P. Roache (1972). Computational Fluid Dynamics, Hermosa Publ., Albuquerque, NM.
- S. Rubin, J. Harris, ed. (1975). Numerical studies of incompressible viscous flow in a driven cavity, NASA SP-378.
- J. C. Strikwerda (1982). An iterative method for solving the Stokes equations, to appear.
- T. D. Taylor, E. Ndefo (1970). Computation of viscous flow in a channel by the method of splitting, Proc. Second Int. Conf. Numer. Methods Fluid Dynamics, p. 356-364.
- R. Temam (1979). Navier-Stokes Equations, North Holland Publ. Co., Amsterdam.
- N. N. Yanenko (1971). The Method of Fractional Steps; the Solution of Problems of Mathematical Physics in Several Variables. Translated by M. Holt, Springer-Verlag, Berlin.

TABLE I

<u>N</u>	<u>err(u)</u>	<u>err(p)</u>	δ_{-h}	<u>sym(u)</u>	<u>sym(p)</u>
20	.35(-3)	.17(-2)	-.44(-5)	.68(-3)	.13(-2)
30	.11(-3)	.86(-3)	-.89(-6)	.22(-3)	.37(-3)
40	.41(-4)	.51(-3)	-.53(-6)	.82(-4)	.15(-3)
60	.19(-4)	.23(-3)	-.50(-7)	.37(-4)	.52(-4)

Errors for test problem 1 for grids with $N + 1$ points on a side for four values of N . The numbers in parenthesis are the decimal exponents i.e. $-35(-3) = .35 \times 10^{-3}$.

TABLE II

<u>N</u>	<u>err(u)</u>	<u>err(p)</u>	δ_{-h}
20	.10(-3)	.21(-2)	-.24(-3)
30	.45(-4)	.92(-3)	-.12(-3)
40	.25(-4)	.48(-3)	-.74(-4)
60	.11(-4)	.22(-3)	-.35(-4)

Errors for test problem 2.

Table III

<u>N</u>	<u>err(u)</u>	<u>err(p)</u>	δ_{-h}
20	.75(-1)	.93(-1)	-.33(-2)
30	.33(-1)	.34(-1)	-.53(-3)
40	.19(-1)	.18(-1)	-.15(-3)
60	.83(-2)	.75(-2)	-.27(-4)

Errors for test problem 3.

TABLE IV

N_1/N_2		1	2	3
		—	—	—
30/20	u	2.8	2.0	2.0
	p	1.7	2.0	2.5
	δ_h	4.0	1.7	4.5
40/30	u	3.4	2.0	1.9
	p	1.8	2.3	2.2
	δ_h	1.9	1.7	4.4
40/20	u	3.1	2.0	2.0
	p	1.7	2.1	2.4
	δ_h	3.1	1.7	4.5
60/30	u	2.5	2.0	2.0
	p	1.9	2.1	2.2
	δ_h	4.2	1.8	4.3
60/40	u	1.9	2.0	2.0
	p	2.0	1.9	2.2
	δ_h	5.8	1.8	4.2

Computed order of accuracy for u, p, and δ_h for the test problems.

A THREE-DIMENSIONAL NUMERICAL MODEL
OF COASTAL, ESTUARINE, AND LAKE CURRENTS

Y. P. Sheng
Aeronautical Research Associates of Princeton, Inc.
P.O. Box 2229, Princeton, New Jersey 08540

H. L. Butler
U.S. Army Engineer Waterways Experiment Station
P.O. Box 631, Vicksburg, Mississippi 39180

ABSTRACT. A mathematical model capable of simulating the three-dimensional, time-dependent currents in coastal, estuaries, and lake waters is presented. Special computational features included in the model are: (1) a time-splitting technique which separates the computation of the slowly varying internal mode (three-dimensional variables) from the computation of the fast-varying external mode (water level and vertically-integrated velocities), (2) an ADI algorithm for the computation of the external mode, (3) an implicit algorithm for the vertical diffusion terms of the internal mode equations, (4) a vertically-stretched coordinate that allows the same order of accuracy in the vertical direction at all horizontal locations, and (5) an algebraically-stretched grid in the horizontal directions. These computational features lead to an efficient and versatile three-dimensional model suitable for long-term simulations. Physical aspects of the model are also discussed. Applications of the model to simulate laboratory flow, tidal currents in an open bight where an analytical solution is available, wind-driven lake currents, and tide-driven and wind-driven coastal currents are also presented.

1. **INTRODUCTION.** The increasing human activities such as dredging and energy production in coastal waters, combined with the increasing concern over the environmental impact of these activities, has created a pressing need for more quantitative understanding of the complex physical processes in coastal waters. Mathematical models, in conjunction with field measurements, can be

used to study many problems of practical interest — such as storm surge prediction, sediment transport and resuspension, dredged material movement, wave prediction, pollutant dispersal, and forces on pipelines. Various mathematical models have been developed to study the hydrodynamic processes of large bodies of water including coastal waters, estuaries, and large lakes. It is fair to say that, for relatively complex mathematical models, resolving the numerical problems is as important and difficult as resolving the physical processes. This work discusses both aspects and emphasizes the numerical aspect.

There exist various time and length scales in the hydrodynamic processes of large bodies of water, ranging from the small scales of the surface waves ($1 \text{ sec} < T < 20 \text{ sec}$, $1 \text{ cm} < L < 500 \text{ m}$) and the mesoscales corresponding to the internal and inertial waves ($N^{-1} < T < f^{-1}$, $100 \text{ m} < L < 100 \text{ km}$), to the large scales associated with the long waves (tides, storm surges, and seiches). Due to a lack of physical understanding and the limitation of computer resources, most existing numerical models of large scale processes do not resolve the small scale and the mesoscale range, but resort to parameterizing the processes in these ranges. For coastal applications, the present model attempts to resolve motions (1) at longer periods than the tidal periods, but less than a month and hence are related to atmospheric forcing (wind stress or curl of wind stress) or river runoffs, (2) at tidal periods and their harmonics and hence are related to tidal forcing and resonance effect of the basin, and (3) at shorter periods than the tidal periods and hence are related to barotropic or baroclinic waves propagating towards the coast. However, current emphasis is placed on the first two cases. Effects of earth rotation, stratification, and bottom topography are included. For barotropic or baroclinic waves propagating in stratified fluid, the wave length may be quite short and thus requires a very fine horizontal grid resolution ($< 1 \text{ km}$). The primary purpose of the current modeling effort is to study the dispersion of particulate or dissolved species (e.g. sediment, dredged material, heat) associated with wind events (on the order of 1 to 10 days) or tidal events (from spring to neap tides).

Due to a limitation of computer resources, simpler models such as vertically-averaged models (Leendertse, 1970; Butler, 1980a) and laterally-averaged models (Blumberg, 1977; Edinger and Buchak, 1979) have been constructed to allow for long-term simulation, at the expense of spatial resolution in one or more dimensions. Such models, although useful in parametric studies and limiting cases such as storm surge predictions, are insufficient for studies of the generally three-dimensional hydrodynamic processes such as sediment transport and wind-driven currents on the continental shelf.

Generally, there are three types of three-dimensional hydrodynamic models: (1) Steady-state models (Gedney and Lick, 1972; Sheng and Lick, 1972; Sheng, 1975) which neglect the transient effect altogether; (2) Rigid-lid, time-dependent models (Bennett, 1977; Sheng, 1975) which eliminate surface gravity waves from the problem; and (3) Free-surface, time-dependent models (Cheng, et al., 1975; Leendertse and Liu, 1975; Sheng, 1975; Forristal, et al., 1977; and Sheng, et al., 1978) which are more general.

In order to study the dynamic response of coastal waters (e.g., the Mississippi Sound and adjacent continental shelf waters) to tides, winds, and meteorological forcing, a three-dimensional, free-surface, time-dependent model is often desired. In addition, the response of coastal waters is strongly influenced by climate, geomorphology, and stratification. Hence, these features have to be properly resolved by the mathematical model. Unfortunately, most three-dimensional, free-surface models require an exceedingly small time step (associated with the propagation of gravity wave over the distance of a horizontal grid spacing), and hence their applications are limited. For example, Leendertse and Liu (1975) used time steps on the order of 10 seconds while applying their model to Chesapeake Bay and San Francisco Bay. Consequently, despite the comprehensiveness of their model, simulation runs were only carried out to a few tidal cycles. Recently, Sheng, et al. (1978) separated the computation of three-dimensional velocities (internal mode) which are governed by slower internal dynamics, from the

computation of water level and mass fluxes (external mode) which are governed by fast surface waves — thus resulting in an efficient three-dimensional model. By computing the internal mode with a fairly large time step ($\sim 1/2$ hr for Lake Erie with a $1/2$ mile grid), the computational efficiency of the three-dimensional model has become comparable to that of a three-dimensional, rigid-lid model or a vertically-averaged model. More recently (Sheng, 1981) we have implemented an implicit numerical scheme for the external computation, thus further increasing the efficiency of the three-dimensional model, and making such a model an attractive operational tool for long-term simulations. Due to the implicit scheme for the external computation, however, a new mode-splitting scheme different from the earlier version (Sheng and Lick, 1980) was designed. Various aspects of the new three-dimensional model are described in the following.

2. GOVERNING EQUATIONS. The basic equations describing the large-scale motion in a large body of water consist of a continuity equation, momentum equations, conservation equations of heat and salinity, and an equation of state. Inherent assumptions are: (1) pressure distribution is hydrostatic in the vertical direction, (2) Boussinesq approximation is valid, and (3) eddy viscosities and diffusivities are used to describe the turbulence. The resulting equations are as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} = & - \left(\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} \right) + fv - \frac{1}{\rho} \frac{\partial p}{\partial x} \\ & + \frac{\partial}{\partial x} \left(A_H \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(A_H \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial z} \left(A_V \frac{\partial u}{\partial z} \right) \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{\partial v}{\partial t} = & - \left(\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial vw}{\partial z} \right) - fu - \frac{1}{\rho} \frac{\partial p}{\partial y} \\ & + \frac{\partial}{\partial x} \left(A_H \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(A_H \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial z} \left(A_V \frac{\partial v}{\partial z} \right) \end{aligned} \quad (3)$$

$$\frac{\partial p}{\partial z} = - \rho g \quad (4)$$

$$\begin{aligned} \frac{\partial T}{\partial t} = & - \frac{\partial uT}{\partial x} + \frac{\partial vT}{\partial y} + \frac{\partial wT}{\partial z} \\ & + \frac{\partial}{\partial x} \left(K_H \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_H \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_V \frac{\partial T}{\partial z} \right) \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{\partial S}{\partial t} = & - \frac{\partial uS}{\partial x} + \frac{\partial vS}{\partial y} + \frac{\partial wS}{\partial z} \\ & + \frac{\partial}{\partial x} \left(D_H \frac{\partial S}{\partial x} \right) + \frac{\partial}{\partial y} \left(D_H \frac{\partial S}{\partial y} \right) + \frac{\partial}{\partial z} \left(D_V \frac{\partial S}{\partial z} \right) \end{aligned} \quad (6)$$

$$\rho = \rho(T, S) \quad (7)$$

where x and y are the horizontal coordinates; z is the vertical coordinate pointing vertically upward to form a right-handed coordinate system with x and y (Figure 1); u , v , and w are the three-dimensional velocities in the x , y , and z directions; t is time; f is the Coriolis parameters; g is the gravitational acceleration; p is the pressure; ρ is the density; T is the

temperature; S is the salinity; A_H , K_H , and D_H are the horizontal eddy coefficients; and A_V , K_V , and D_V are the vertical eddy coefficients.

At the free surface, the appropriate boundary conditions are: (a) the wind stress is specified,

$$A_V \frac{\partial u}{\partial z} = \tau_{sx} , \quad A_V \frac{\partial v}{\partial z} = \tau_{sy} \quad (8)$$

where τ_{sx} and τ_{sy} are the wind stresses in the x and y directions respectively and are functions of the wind velocity at some height; (b) the kinematic condition is satisfied,

$$w = \frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y} \quad (9)$$

where ζ is the elevation of the free surface; (c) the dynamic condition is satisfied,

$$p = p_a \quad (10)$$

where p_a is the atmospheric pressure, and (d) the heat flux and salt flux are specified,

$$K_V \frac{\partial T}{\partial z} = q_s = \bar{h} (T - T_e); \quad \frac{\partial S}{\partial z} = 0 \quad (11)$$

where T_e is the equilibrium air temperature at which the surface heat flux is zero and \bar{h} is the surface heat transfer coefficient.

At the bottom, the boundary conditions are: (a) a quadratic stress law is valid:

$$\underline{\tau}_b = \rho C_D \underline{v}_b v_b \quad (12)$$

where $\underline{\tau}_b$ is the bottom shear stress vector, ρ is the water density, C_D is the skin-friction coefficient, v_b is the magnitude of the bottom current, while \underline{v}_b is the bottom current vector; (b) the heat flux or temperature is specified,

$$k_V \frac{\partial T}{\partial z} = q_b \quad \text{or} \quad T = T_b \quad (13)$$

and (c) the salt flux is specified,

$$\frac{\partial S}{\partial z} = 0 \quad (14)$$

In the numerical model, the above equations and boundary conditions are actually solved in non-dimensional form. In addition, anticipating significant variation of bottom topography in the horizontal direction, the x, y, z coordinate system is vertically-stretched to a x, y, σ coordinate system, such that an equal number of grid points exist in the shallow coastal and the deep offshore areas (Figure 2). The transformation takes the form:

$$\sigma = z/h(x, y) \quad (15)$$

where $h(x, y)$ is the local water depth of the basin. Such a transformation leads to the same order of numerical accuracy in the vertical direction at all horizontal locations. Variable grid spacing may be used in the vertical direction to allow finer resolution within boundary layers, e.g., the bottom boundary layer and the thermocline.

To better resolve the complex shoreline geometries and bottom features, a non-uniform grid is often required in the x and y directions (Butler and Sheng, 1982). This non-uniform grid (x,y,z) is further mapped into a uniform grid (α,γ,σ) :

$$\begin{aligned}x &= a_x + b_x \alpha^c \\y &= a_y + b_y \gamma^c\end{aligned}\tag{16}$$

The resulting equations and boundary conditions in α,γ,σ grid system in non-dimensional form are presented in the Appendix.

The system of equations would admit surface gravity waves, internal waves, inertial waves, and Rossby waves (if β -plane approximation is used). Various time and spatial scales may exist in the numerical solution, depending on the grid resolution, the forcing function, and the location. The various time scales in an enclosed basin have been studied extensively by Haq, Lick and Sheng (1974).

2.1 MODE SPLITTING. In the present study, numerical computation of the three-dimensional variables (internal mode), which are governed by slower dynamics, are separated from the computation of the vertically-integrated variables (external mode). This so-called "mode splitting" technique resulted in significant improvement of the numerical efficiency of a three-dimensional hydrodynamic model for Lake Erie (Sheng et al., 1978) and was detailed in Sheng and Lick (1980). Basically, it allows for computation of the three-dimensional flow structures with minimal additional cost over computation of the two-dimensional flow with a vertically-integrated model.

2.2 EXTERNAL MODE. The external mode, as described by the water level (ζ) and the vertically-integrated mass fluxes (U and V), is governed by the following equations:

$$\frac{\partial \zeta}{\partial t} + \beta \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \right) = 0\tag{17}$$

$$\begin{aligned}
\frac{\partial U}{\partial t} = & -\frac{h}{\mu_x} \frac{\partial \zeta}{\partial x} + V + \tau_{sx} - \tau_{bx} + E_H \left[\frac{1}{\mu_x} \frac{\partial}{\partial x} \left(\frac{1}{\mu_x} \frac{\partial U}{\partial x} \right) + \frac{1}{\mu_y} \frac{\partial}{\partial y} \left(\frac{1}{\mu_y} \frac{\partial U}{\partial y} \right) \right] \\
& - \frac{E_H}{h} \left[\frac{1}{\mu_x} \frac{\partial}{\partial x} \left(\frac{1}{\mu_x} \frac{\partial h}{\partial x} \right) + \frac{1}{\mu_y} \frac{\partial}{\partial y} \left(\frac{1}{\mu_y} \frac{\partial h}{\partial y} \right) \right] \left(\frac{\partial U}{\partial \sigma} \right)_{\sigma = -1} \\
& - R_B \left[\frac{1}{\mu_x} \frac{\partial}{\partial x} \left(\frac{U^2}{h} \right) + \frac{1}{\mu_y} \frac{\partial}{\partial y} \left(\frac{UV}{h} \right) + h(u\omega) \right]_{\sigma = 0} = -\frac{h}{\mu_x} \frac{\partial \zeta}{\partial x} + U_x \quad (18)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial V}{\partial t} = & -\frac{h}{\mu_y} \frac{\partial \zeta}{\partial y} - U + \tau_{sy} - \tau_{by} + E_H \left[\frac{1}{\mu_x} \frac{\partial}{\partial x} \left(\frac{1}{\mu_x} \frac{\partial V}{\partial x} \right) + \frac{1}{\mu_y} \frac{\partial}{\partial y} \left(\frac{1}{\mu_y} \frac{\partial V}{\partial y} \right) \right] \\
& - \frac{E_H}{h} \left[\frac{1}{\mu_x} \frac{\partial}{\partial x} \left(\frac{1}{\mu_x} \frac{\partial h}{\partial x} \right) + \frac{1}{\mu_y} \frac{\partial}{\partial y} \left(\frac{1}{\mu_y} \frac{\partial h}{\partial y} \right) \right] \left(\frac{\partial V}{\partial \sigma} \right)_{\sigma = -1} \\
& - R_B \left[\frac{1}{\mu_x} \frac{\partial}{\partial x} \left(\frac{UV}{h} \right) + \frac{1}{\mu_y} \frac{\partial}{\partial y} \left(\frac{V^2}{h} \right) + h(v\omega) \right]_{\sigma = 0} = -\frac{h}{\mu_y} \frac{\partial \zeta}{\partial y} + V_y \quad (19)
\end{aligned}$$

where ω is the vertical velocity in the x, y, σ system as defined by (A.11), R_B is the Rossby number and E_H is the horizontal Ekman number as defined in the Appendix, τ_{sx} and τ_{sy} are shear stresses at the free surface, while τ_{bx} and τ_{by} are the bottom shear stresses which are computed from the three-dimensional velocity profiles from Equation (A.13), where \underline{v}_b is the horizontal velocity vector evaluated at a point z_+ above the bottom within the logarithmic layer. Ideally, the drag coefficient C_D is a function of the bottom roughness (z_0), the distance above the bottom (z_+), and the stability of the flow near the bottom (Sheng, 1980):

$$C_D = \left[\frac{K}{2\ln(z_+/z_0) + \phi} \right]^2 \quad (20)$$

where K is the von-Karman constant and ϕ is a stability function (Businger, et al., 1971; Lewellen and Sheng, 1980.)

Sternberg (1972) measured the steady-state flow over a variety of bottom conditions in both the laboratory and the ocean, and found C_D to be generally in the neighborhood of 0.004. However, recent studies have found that C_D in the ocean, particularly in the presence of wind waves, may be an order of magnitude higher or more (Grant, 1981). Recent study of the bottom boundary layer under current and wave interaction by us using a second-order closure turbulence model (Sheng and Lewellen, 1982) quantitatively confirmed this fact. Studies of tidal currents in a shallow estuary also revealed that C_D is on the order of 0.035 (Brown and Trask, 1980).

Bottom friction as represented by Equation (20) allows one to include the effect of oscillating wave-induced current on the mean current in the hydrodynamic model, and is believed to be physically more meaningful than the Chezy type formula used in a conventional vertically-averaged model.

The vertically-averaged model gives results similar to the external mode of the three-dimensional model when flow is rather homogeneous in the vertical direction. However, due to its failure to resolve the vertical Ekman layer and the vertical stratification, it may yield quite different results when two-layer flow or stratification exists.

2.3 INTERNAL MODE. The internal mode of the flow is described by the three-dimensional velocities (u , v , w), temperature (T), salinity (S), and density (ρ). Equations for T , S , ρ as shown by Equations (A.6), (A.7) and (A.8) are solved along with two equations for the perturbation velocities $u' \equiv u - U/h$ and $v' \equiv v - V/h$:

$$\frac{\partial u'}{\partial t} = B_x - \frac{D_x}{h} + \frac{E_v}{h^2} \frac{\partial}{\partial \sigma} \left[A_v \frac{\partial}{\partial \sigma} \left(u' + \frac{U}{h} \right) \right] \quad (21)$$

$$\frac{\partial v'}{\partial t} = B_y - \frac{D_y}{h} + \frac{E_v}{h^2} \frac{\partial}{\partial \sigma} \left[A_v \frac{\partial}{\partial \sigma} (v' + \frac{v}{h}) \right] \quad (22)$$

where B_x and B_y are defined in Equations (A.3) and (A.4), respectively. These equations are obtained by subtracting the vertically-integrated equations, Equations (18) and (19), from the u - and v - equations, Equations (A.3) and (A.4), and hence do not contain the pressure gradient terms. The computation of these equations are thus not limited by the stringent numerical time step associated with the fast surface gravity waves.

However, the mode-splitting technique used in this study is somewhat different from the one used in Sheng et al. (1978). In that study, the governing equations for the internal flow variables consisted of equations of motion and the continuity equation in terms of the differences of velocities between adjacent grid points in the vertical direction. As mentioned earlier, this is due to the fact that an explicit scheme was used for the external mode in Sheng et al. (1978), while an implicit scheme is used here.

3. TURBULENCE PARAMETERIZATION. Various levels of turbulence parameterization, from the simple constant eddy viscosity model (Gedney and Lick, 1972; Forristall, et al., 1977) to the second-order closure model of turbulence (Sheng and Lewellen, 1982), have been used in hydrodynamic models. In the present study, a semi-empirical theory of vertical mixing is used. The effect of stratification, as measured by the Richardson number, Ri , on the intensity of vertical turbulent mixing is parameterized by an empirical stability function:

$$Ri = \frac{-\frac{g}{\rho} \frac{\partial \rho}{\partial z}}{\left(\frac{\partial u}{\partial z} \right)^2} \quad (23)$$

$$A_v = A_{v0} (1 + \sigma_1 Ri)^{m_1} \quad (24)$$

$$K_v = K_{v0} (1 + \sigma_2 Ri)^{m_2} \quad (25)$$

$$D_v = D_{v0} (1 + \sigma_3 R_1)^{m_3} \quad (26)$$

where A_{v0} , K_{v0} , and D_{v0} are the eddy coefficients in the absence of any density stratification and σ_1 , σ_2 , σ_3 , m_1 , m_2 , and m_3 are empirically determined constants. As shown in Figure 3(a), great discrepancy exists among the various forms of the stability functions determined empirically by various workers (Munk and Anderson, 1948; Bowden and Hamilton, 1975; Blumberg, 1977). In addition, the critical Richardson numbers, at which the turbulence is completely damped by buoyancy, given by these formulas are much too high (-10) compared to the measured value of 0.25 (Erikson, 1978; Davis et al. 1981). Much of the discrepancy among various formulae probably resulted from the difference in numerical schemes used and the different nature of the water bodies studied. To unify this discrepancy, it is believed that stability functions determined from a second-order closure model of turbulence should be used. Donaldson (1973) compared the second-order closure model with the K-theory of turbulence and obtained the stability functions by assuming a balance between turbulence production and dissipation, i.e., the so-called "super-equilibrium" condition. As shown in Figure 3(b), such a stability function leads to a critical Richardson number much closer to 0.25. In order to utilize these relationships, a turbulence length scale, Λ , has to be defined empirically.

4. GRID STRUCTURE. The basic staggered-grid structure used in the study is shown in Figure 4. The indices i , j , and k correspond respectively to x , y , and z coordinates. In the x - y plane, the variables u and U are computed at the mid-points of the two boundaries parallel to the y -axis, v and V are computed at the mid-points of the two boundaries parallel to the x -axis, and w and ϵ are computed at the center of the grid. Shorelines are fitted with a rectangular grid such that u and U are either zero or prescribed by river flows along a shoreline parallel to the y -axis, and v and V are zero or prescribed by river flows along a shoreline parallel to the x -axis. In the vertical direction, the free surface and the bottom both fall on the full grid points on which the w 's and the shear stresses are either computed or prescribed.

5. NUMERICAL ALGORITHMS.

5.1 EXTERNAL MODE ALGORITHMS. Treating all the terms on the left hand side of the vertically-integrated equations (Equations 18 and 19) implicitly, the following finite difference equations in matrix form are obtained:

$$(1 + \lambda_x + \lambda_y) W^{n+1} = W^n + \Delta t \cdot D^n \quad (27)$$

where

$$\lambda_x = \frac{A \Delta t}{\Delta x} \delta_x \quad \lambda_y = \frac{B \Delta t}{\Delta y} \delta_y \quad (28)$$

and

$$\begin{aligned} A &= \begin{pmatrix} 0 & B/\mu_x & 0 \\ h/\mu_x & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & B &= \begin{pmatrix} 0 & 0 & B/\mu_y \\ 0 & 0 & 0 \\ h/\mu_y & 0 & 0 \end{pmatrix} \\ D &= \begin{pmatrix} 0 \\ D_x \\ D_y \end{pmatrix} & W &= \begin{pmatrix} \zeta \\ U \\ V \end{pmatrix} \end{aligned} \quad (29)$$

where Δx and Δy are the spatial grid spacings in x and y directions, Δt is the time step, and the superscripts $n + 1$ and n indicate the present and previous time step of integration. Rewriting Equation (27) and neglecting terms of $O(\Delta t^2)$ yields the following equations:

$$(1 + \lambda_x) W^* = (1 - \lambda_y) W^n + \Delta t \cdot D^n \quad (30)$$

$$(1 + \lambda_y) W^{n+1} = W^* + \lambda_y W^n \quad (31)$$

These two equations can be solved consecutively in an efficient manner by inversion of tridiagonal matrices in the x-direction (x-sweep) and y-direction (y-sweep). Furthermore, only ζ and U are solved in the x-sweep, while only ζ and V are solved in the y-sweep. This results in a significant saving than when solving the original Equation (27) with iterative or direct method. Using this implicit scheme, the Courant number based on the wave speed $[(gh_{\max})^{0.5} \Delta t / \Delta x]$ could be as large as 100, compared to only 1 for the explicit method. In reality, however, the Courant numbers for our simulation have generally been between 5 and 10 to maintain sufficient numerical accuracy. For a detailed discussion on the ADI method for vertically-integrated shallow water equations, one is referred to another paper in these proceedings (Butler and Sheng, 1982).

5.2 INTERNAL MODE ALGORITHMS. Due to the extreme shallowness of water in the coastal area (-2m), the numerical stability associated with the vertical diffusion term imposes a very small time step (-20 sec) if a forward-time, central-space (FTCS) scheme is used. To alleviate this problem, three numerical schemes were tested: (1) Du-Fort Frankel Scheme, (2) Hopskotch Scheme, and (3) fully implicit scheme. The fully implicit scheme is implemented in this model:

$$u^{n+1} = u^n + \Delta t (A_x^n + B_x^n) + \frac{\Delta t}{h^2} \frac{\partial}{\partial \sigma} \left[A_v \frac{\partial}{\partial \sigma} (u^{n+1} + u^{n+1}/h) \right] \quad (32)$$

$$v^{n+1} = v^n + \Delta t (A_y^n + B_y^n) + \frac{\Delta t}{h^2} \frac{\partial}{\partial \sigma} \left[A_v \frac{\partial}{\partial \sigma} (v^{n+1} + v^{n+1}/h) \right] \quad (33)$$

The bottom friction terms in both equations are also treated implicitly. However, care must be taken to ensure that the vertically-integrated perturbation velocities at each horizontal location (i,j) always equal zero:

$$\sum_{k=1}^{K_{\max}} u'_{i,j,k} = 0, \quad \sum_{k=1}^{K_{\max}} v'_{i,j,k} = 0 \quad (34)$$

After the equations for u' and v' are solved, equations for salinity, temperature, and density may then be solved with the same two time level numerical scheme and implicit vertical diffusion treatment.

A two time level scheme is preferred in this study for the following reasons: (1) reduction of storage required on computer resource (VAX 11/780), and (2) alleviation of the time-splitting phenomenon that may otherwise result from a three time level numerical scheme.

A flow chart of the solution algorithm is shown in Figure 5. The model may be run exclusively in the external mode with a fairly large time step. The internal mode may be updated every so often as desired or as dictated by the physical problem of interest. For a detailed discussion on this subject and a somewhat different treatment of the internal mode, one is referred to Sheng and Lick (1980). It should be pointed out that the maximum allowable time steps, for both the external and the internal modes, are now limited by the ratio between the horizontal grid spacing and the maximum advection speed. These time steps are generally one to two orders of magnitude larger than the time step for a simple three-dimensional, free-surface model as imposed by the propagation of surface gravity waves.

To remove numerical noise in the form of short-wave oscillation (with wave length typically twice the grid spacing) which may lead to numerical instability (Orszag and Israeli, 1974), a spatial filter was designed (Sheng et al., 1978). Such a filter will remove the undesirable numerical oscillations when identified, but leave the rest of the transient solution

intact. For most of our coastal applications so far, there was little need to use the spatial filter.

6. APPLICATIONS.

6.1 SIMULATION OF VICKSBURG TIDAL FLUME. The three-dimensional model was used to simulate the flow measured in Test 29 of Ippen and Harlemans' (1961) experiments in the Vicksburg tidal flume. The rectangular flume was 100 m long, 22.86 cm wide, and 15.24 cm deep, with one closed end and one open end connected to a large reservoir where sinusoidal tides in the form of $\zeta = \zeta_0 \cos(2\pi t/T)$, $\zeta_0 = 1.52$ cm and $T = 600$ sec, were generated. Using $A_v = 1$ cm²/sec, $C_0 = 0.004$, $\Delta x = 5$ m, $\Delta y = 4.56$ cm, and $\Delta z = 0.3$ cm, our results agree well with the measurements, even when $\Delta t = 30$ sec (Courant number = 8) was used (Figure 6).

6.2 TIDAL FLOW IN AN OPEN BIGHT. The vertically-integrated version of our model was used to simulate the tidal flow in a square bight as computed analytically by van de Kreeke and Chiu (1980). Consider a square basin of constant depth of 10 m and length of 150 km, with vertical walls along the West and the South, while open boundaries exist along the East and the North. The water level at the open boundaries is selected such that the resulting solutions consist of the sum of two progressive waves propagating in the East-West direction and two progressive waves propagating in the North-South direction. The tidal amplitude at the Northeast corner is assumed to be 30 cm with a period of 24 hours.

Using $\Delta t = 2$ hours and $\Delta x = 15$ km in our model, the computed water level and velocity field agree quite well with their analytical results. Figures 7 and 8 represent the water level and velocity field at the instant when ζ is 0 at the Northeast corner while ζ is maximum at the Southwest corner. Considering the fact that the Courant number is 8 in this simulation, the agreement between model results and analytical results is indeed quite reasonable.

6.3 TIME-DEPENDENT CURRENTS IN A LAKE. To demonstrate the effect of various bottom boundary conditions on the computed water-level and currents, let us consider a 50 km square lake with a linearly increasing bottom from 2.5 m to 7.5 m. An impulsive wind stress of $\tau_x = 1 \text{ dyne/cm}^2$ was applied at $t = 0$ along the direction of bottom contours. Numerical results were obtained for three conditions: (1) three-dimensional model with a no-slip boundary condition, i.e., $u = v = 0$ at the bottom, (2) three-dimensional model with a quadratic stress law ($C_D = 0.004$) and (3) vertically-integrated model with the same quadratic stress law.

Results at selected locations are shown in Figure 9. Water level predicted with the no-slip condition exhibited a much faster decay time and much stronger bottom dissipation. Hence at the steady state, a stronger set-up is required to balance the wind stress and the bottom stress. As shown in Figure 8, vertical velocity structure due to the no-slip condition also differs considerably from that due to the quadratic stress law. The quadratic stress law yields a flatter velocity profile near the bottom, resembling a turbulent boundary layer over a flat bottom. The no-slip condition, on the other hand, yields a parabolic profile near the bottom resembling a laminar boundary layer.

The large difference in the near-bottom currents as computed by the no-slip condition vs. the quadratic stress law is of particular significance if one's primary interest is in the transport of pollutants in shallow waters, where the bottom boundary layer plays a dominant role. The no-slip condition should not be used unless an extremely fine grid is used to resolve the laminar sublayer.

The vertically-integrated model produced a water level quite comparable to the three-dimensional model. However, the vertically-integrated model cannot resolve the vertical velocity structure as shown in Figure 10.

Various applications of the three-dimensional model to Lake Erie have been reported before (e.g., Sheng and Lick, 1972; Sheng et al., 1978; Sheng 1980). A rather interesting example is given here to illustrate the

steady-state response of coastal waters to wind forcing both in the absence of and in the presence of a proposed jetport in the lake (Figure 11). It is clear from these simulations that the coastal currents are not significantly affected by the presence of a jetport island. However, in the presence of a jetport island and a causeway to the shore, the coastal currents are appreciably modified.

6.4 TIDAL CURRENTS IN MISSISSIPPI SOUND

AND ADJACENT SHELF WATERS. The three-dimensional model has been applied to simulate the tidal currents off the Mississippi Coast in an idealized grid (51 x 51 x 5, with $\Delta x = \Delta y = 3$ km and $\Delta \sigma = 1/5$), as shown in Figure 12.

Barrier islands (Dauphin, Petit Bois, Horn, Ship, and Chandelier) are approximately represented by the solid line barriers within the grid. Open boundary extends along the South ($x = L$) and the East ($y = L$). Initially, the entire basin is assumed to be quiescent with $\zeta = 0$ everywhere. Flow is forced by the following boundary condition along the open boundaries:

$$\zeta = \zeta_0 \sin \left[\frac{2\pi t}{T} - \phi(x) \right] \quad (35)$$

where $\phi(x)$ is computed from

$$\phi(x) = (L-x) / \sqrt{gH_{avg}} \quad (36)$$

where H_{avg} is the average depth between x and $x = L$ along the open boundary. ζ_0 is assumed to be 30 cm and T is taken as 24 hrs.

The tidal currents over the entire basin at the end of a 4-day simulation (flood tide) are shown in Figure 13. The near-surface currents ($\sigma = -0.1$) near the Mississippi Sound are much weaker than those in the open shelf waters. However, near the bottom ($\sigma = -0.9$), the currents over the entire basin are quite comparable in magnitude.

To closely examine the tidal currents within the Mississippi Sound, we have shown in Figures 14 and 15 the detailed currents at two stages during the flood tide within a narrow coastal strip including the Sound. It is apparent that modest currents exist within the major passes except to the west of Ship Island. Based on our computation, the average flow through the passes is about 1.8×10^6 CFS, which is very close to the estimate by Escoffier (1978) based on measured tidal currents. It is interesting to note that the maximum bottom shear stress occurs within the major passes. During spring tides, these strong stresses may cause resuspension of sediments in these areas. For a more detailed description of this application, one is referred to Sheng (1981).

6.5 WIND-DRIVEN CURRENTS OFF THE MISSISSIPPI COAST. Strong winds frequently exist in the study area. The winds in the Gulf of Mexico are predominantly from the North in winter and South to Southeast in summer. However, winds are strongest in winter from the West and the Northwest. Currents and water level induced by the strong winds can be much greater than those induced by tides and hence are of primary interest to us.

As an example, we present the response of the coastal waters under an impulsive wind stress of 3 dyne/cm^2 from the West. For simplicity, adiabatic boundary conditions with zero surface elevation are applied along the open boundaries. This eliminates the effect of shelf waves on the circulation, but allows simulation of response of study area to local wind forcing. To study the effect of Loop Currents on the coastal circulation, our limited-area model should be coupled with a larger model which includes the entire Gulf.

The response of the coastal waters to the wind forcing is illustrated by the time variation of the bottom currents at three locations (Figure 16). Within the Sound (location B), the local flow has reached a steady state within an inertial period. At an offshore location (C), the response is somewhat slower. At location D near the open boundary, the response is even slower as manifested by the distinctive inertial period in its oscillating bottom current.

The mass fluxes, near-surface currents, near-bottom currents, and bottom shear stresses caused by the westerly wind are shown in Figure 17. Within the Sound, the local geometry and bottom topography play the important roles in causing a predominantly alongshore flow in the direction of the wind. In such limiting cases, a vertical 2-D (x-z) model may be used for parametric studies associated with the navigation channels within the Sound.

According to a laboratory flume study on the erodibility of the Mississippi Sound sediments (Sheng, 1981), it is expected that the bottom shear stress generated by the strong westerly wind in winter will cause appreciable resuspension of the sediments. The exchange of water masses between the Sound and adjacent offshore waters may result in transport of sediments into or out of the Sound.

Open boundary conditions for a limited-area coastal circulation model remain to be an unresolved challenge. Boundary conditions along the open boundaries may be provided from a larger model with dynamic coupling between the two models (e.g., Sheng, 1975.) For our application to the Mississippi coastal waters, tidal constituents from a tide model (Reid and Whitaker, 1982) for the entire Gulf of Mexico will be used as boundary conditions. Resonance effect due to the basin may also be included by using the mass fluxes, in addition to the surface elevation, as boundary conditions. To allow disturbances to be propagated into and out of the coastal area without being reflected back into the area, a modified radiation boundary condition is being developed. Its successful application to idealized and practical problems will be reported in a forthcoming paper by us.

7. CONCLUSION. We have presented the detailed formulation of a three-dimensional numerical model which is capable of realistically describing the short- and long-term, time-dependent currents in coastal, estuarine, and lake waters.

Coordinate stretching was applied to the spatial numerical grids in both the vertical and the horizontal directions to allow for flexibility and accuracy in resolving complex geometrical and topographical features. The

governing equations and boundary conditions were solved in the transformed coordinates, where the horizontal grids are always uniformly spaced, while the vertical grid may be non-uniformly spaced. Special integration techniques were implemented to allow for numerical time step significantly larger than that for conventional three-dimensional hydrodynamic models. Various physical aspects of the model such as turbulence formulation, bottom friction, and open boundary conditions were also discussed. If desired, the model may be run as a two-dimensional, vertically integrated model or a two-dimensional, laterally-averaged model.

Various applications of this numerical model demonstrated the feasibility of applying it to the various projects within the Army Corps of Engineers — such as storm surge prediction, sediment transport, dredged material movement, and maintenance of navigation channels.

8. ACKNOWLEDGEMENT. Recent development and application of the three-dimensional model was supported by the U.S. Army Engineer Waterways Experiment Station under contract DACW 39-80-C-0087. Previous efforts on the three-dimensional model have been supported by the Environmental Protection Agency and the U.S. Army Engineer Waterways Experiment Station.

APPENDIX

Governing Equations and Boundary Conditions

To write the equations and boundary conditions in non-dimensional form, the following non-dimensional quantities are defined:

$$(u^*, v^*, w^*) = (u, v, wL/H)/U_r$$

$$(x^*, y^*, z^*) = (x, y, zL/H)/L$$

$$(\tau_{sx}^*, \tau_{sy}^*) = (\tau_{sx}, \tau_{sy})/p(A_v)_r U_r$$

$$t^* = tf, \quad p^* = p/\rho U_r f L, \quad \tau^* = \rho U_r f L \quad (A.1)$$

$$\rho^* = \rho/\rho_r, \quad T^* = (T - T_r)/T_r, \quad \zeta^* = g\zeta/fU_r L$$

$$A_H^* = A_H/(A_H)_r, \quad K_H^* = K_H/(K_H)_r, \quad D_H^* = D_H/(D_H)_r$$

$$A_V^* = A_V/(A_V)_r, \quad K_V^* = K_V/(K_V)_r, \quad D_V^* = D_V/(D_V)_r$$

where quantities with subscript r are reference quantities, and H and L are vertical and horizontal length scales.

Suppressing the asterisk (*) for clarity, and transforming the (x, y, z) coordinate system to a vertically-stretched (α, γ, σ) system, and further replacing (α, γ) with (x, y) , the equations become:

Continuity

$$\frac{1}{\mu_x} \frac{\partial(hu)}{\partial x} + \frac{1}{\mu_y} \frac{\partial(hv)}{\partial y} + h \frac{\partial w}{\partial \sigma} = 0 \quad (A.2)$$

x-momentum

$$\begin{aligned}
 \frac{\partial u}{\partial t} &= -\frac{R_B}{h} \left[\frac{\partial(hu^2)}{\mu_x \partial x} + \frac{\partial(huv)}{\mu_y \partial y} + h \frac{\partial(\omega u)}{\partial \sigma} \right] + v - \frac{\partial \zeta}{\mu_x \partial x} - A_x \\
 &+ \frac{E_H}{h} \left[\frac{\partial}{\mu_x \partial x} \left(h \frac{\partial u}{\mu_x \partial x} \right) + \frac{\partial}{\mu_y \partial y} \left(h \frac{\partial u}{\mu_y \partial y} \right) \right] + \frac{E_V}{h^2} \frac{\partial}{\partial \sigma} \left(A_V \frac{\partial u}{\partial \sigma} \right) \quad (A.3) \\
 &\equiv -\frac{\partial \zeta}{\mu_x \partial x} + B_x + \frac{E_V}{h^2} \frac{\partial}{\partial \sigma} \left(A_V \frac{\partial u}{\partial \sigma} \right)
 \end{aligned}$$

y-momentum

$$\begin{aligned}
 \frac{\partial v}{\partial t} &= -\frac{R_B}{h} \left[\frac{\partial(huv)}{\mu_x \partial x} + \frac{\partial(hv^2)}{\mu_y \partial y} + h \frac{\partial(\omega v)}{\partial \sigma} \right] - u - \frac{\partial \zeta}{\mu_y \partial y} - A_y \\
 &+ \frac{E_H}{h} \left[\frac{\partial}{\mu_x \partial x} \left(h \frac{\partial v}{\mu_x \partial x} \right) + \frac{\partial}{\mu_y \partial y} \left(h \frac{\partial v}{\mu_y \partial y} \right) \right] + \frac{E_V}{h^2} \frac{\partial}{\partial \sigma} \left(A_V \frac{\partial v}{\partial \sigma} \right) \quad (A.4) \\
 &\equiv -\frac{\partial \zeta}{\mu_y \partial y} + B_y + \frac{E_V}{h^2} \frac{\partial}{\partial \sigma} \left(A_V \frac{\partial v}{\partial \sigma} \right)
 \end{aligned}$$

Hydrostatic

$$\frac{\partial p}{\partial \sigma} = - \rho h \frac{R_B}{Fr^2} \quad (A.5)$$

Energy

$$\begin{aligned} \frac{\partial T}{\partial t} = & - \frac{R_B}{h} \left[\frac{\partial(huT)}{\mu_x \partial x} + \frac{\partial(hvT)}{\mu_y \partial y} + h \frac{\partial(\omega T)}{\partial \sigma} \right] + \frac{Ev}{Pr_v} \frac{L}{H} \frac{1}{h^2} \frac{\partial}{\partial \sigma} \left(K_v \frac{\partial T}{\partial \sigma} \right) \\ & + \frac{E_H}{Pr_H} \left[\frac{\partial}{\mu_x \partial x} \left(h \frac{\partial T}{\mu_x \partial x} \right) + \frac{\partial}{\mu_y \partial y} \left(h \frac{\partial T}{\mu_y \partial y} \right) \right] \end{aligned} \quad (A.6)$$

Salt

$$\begin{aligned} \frac{\partial S}{\partial t} = & - \frac{R_B}{h} \left[\frac{\partial(huS)}{\mu_x \partial x} + \frac{\partial(hvS)}{\mu_y \partial y} + h \frac{\partial(\omega S)}{\partial \sigma} \right] + \frac{E_v}{Sc_v} \frac{L}{H} \frac{1}{h^2} \frac{\partial}{\partial \sigma} \left(D_v \frac{\partial S}{\partial \sigma} \right) \\ & + \frac{E_H}{Sc_H} \left[\frac{\partial}{\mu_x \partial x} \left(h \frac{\partial S}{\mu_x \partial x} \right) + \frac{\partial}{\mu_y \partial y} \left(h \frac{\partial S}{\mu_y \partial y} \right) \right] \end{aligned} \quad (A.7)$$

Equation of State

$$\Delta \rho^- = \sigma_t = 10^3 (\rho - 1) = \sum_{ij} a_{ij} T^j S^j \quad (A.8)$$

where A_x and A_y are defined as:

$$A_x \equiv \frac{1}{\mu_x} \frac{R_B}{Fr^2} \left[\int_{\sigma}^0 \frac{\partial \rho}{\partial x} d\sigma + \frac{\partial h}{\partial x} \left(\int_{\sigma}^0 \rho d\sigma + \sigma \rho \right) \right] \quad (A.9)$$

$$A_y \equiv \frac{1}{\mu_y} \frac{R_B}{Fr^2} \left[\int_{\sigma}^0 \frac{\partial \rho}{\partial y} d\sigma + \frac{\partial h}{\partial y} \left(\int_{\sigma}^0 \rho d\sigma + \sigma \rho \right) \right] \quad (A.10)$$

and where

$$\omega = \frac{w}{h} - \frac{\sigma}{h} \left(\frac{u}{\mu_x} \frac{\partial h}{\partial x} + \frac{v}{\mu_y} \frac{\partial h}{\partial y} \right) \quad (A.11)$$

The constants a_{ij} for the equation of state are listed in Sheng (1981). μ_x and μ_y are stretching rates defined as $d\alpha/dx$ and dy/dy , respectively. The dimensionless parameters are defined as: R_B = Rossby number = U_r/fL , E_H = horizontal Ekman number = $(A_H)_r/fL^2$, E_V = vertical Ekman number = $(A_V)_r/fH^2$, Fr = Froude number = $U_r/(gh)^{0.5}$, Pr_H = horizontal Prandtl number = K_H/A_H , Pr_V = vertical Prandtl number = K_V/A_V , Sc_H = horizontal Schmidt number = D_H/A_H , Sc_V = vertical Schmidt number = D_V/A_V , U_r = reference velocity, and the subscript r indicates reference quantities.

The higher order terms in the horizontal diffusion terms contain bottom slopes and/or their products, and hence are generally small compared to the listed leading terms when $H/L \ll 1$. It should be noted that in deriving Equations (A.3) and (A.4), the vertically-integrated form of the hydrostatic equation was substituted into the x- and y-momentum equations. Horizontal eddy coefficients were assumed to be independent of space and time. In general, the vertical eddy coefficients are functions of the wind, local depth, and vertical density gradient.

The appropriate non-dimensional boundary conditions in the vertically-stretched coordinate system are as follows:

Surface, $\sigma = 0$

$$\frac{\partial u}{\partial \sigma} = \frac{h\tau_{sx}}{A_v}, \quad \frac{\partial v}{\partial \sigma} = \frac{h\tau_{sy}}{A_v}, \quad \omega = \frac{1}{\beta h} \frac{\partial \zeta}{\partial t}$$

$$\frac{\partial T}{\partial \sigma} = \frac{Hh\tilde{n}}{K_v} (T - T_e), \quad \frac{\partial S}{\partial \sigma} = 0, \quad \text{where } \beta \equiv \frac{gh}{L^2 f^2} \quad (\text{A.12})$$

Bottom, $\sigma = -1$

$$\tau_b = R_B C_D \underline{V}_b V_b, \quad \frac{1}{h} \frac{\partial T}{\partial \sigma} = q_b \text{ or } T = T_b \quad (\text{A.13})$$

River inflow or outflow

$$u = u(x, y, \sigma), \quad v = v(x, y, \sigma), \quad T = T(x, y, \sigma), \quad S = S(x, y, \sigma) \quad (\text{A.14})$$

Shore

$$u = v = 0, \quad \frac{\partial T}{\partial n} = 0, \quad \frac{\partial S}{\partial n} = 0 \quad (\text{A.15})$$

where \underline{V}_b is the velocity vector at the bottom, while V_b is its magnitude.

REFERENCES

- Bennett, J.R., 1977; "A Three-Dimensional Model of Lake Ontario's Summer Circulation," I. Comparison with Observations, J. Phys. Oceano., 7, pp. 591-601.
- Blumberg, A.F., 1977; "Numerical Mode of Estuarine Circulation," J. Hyd. Div. ASCE, 103, No. HY3.
- Bowden, K.F. and P. Hamilton, 1975; "Some Experiments with a Numerical Model of Circulation and Mixing in a Tidal Estuary," Estuarine Coastal Marine Sci., 3, 281.
- Brown, W.S. and R.P. Trask, 1980; "A Study of Tidal Energy Dissipation and Bottom Stress in an Estuary," J. Phys. Oceano., 10, pp. 1742-1754.
- Businger, J.A., J.C. Wyngaard, Y. Izumi, and E.F. Bradley, 1971; "Flux-Profile Relationships in the Atmospheric Surface Layer," J. Atmos. Sci., 28, pp. 181-189.
- Butler, H.L., 1980; "Evolution of a Numerical Model for Simulating Long-Period Wave Behavior in Ocean-Estuarine Systems," in Estuarine and Wetland Processes (P. Hamilton, ed.), Springer-Verlag, Berlin, Heidelberg, pp. 368-378.
- Butler, H.L. and Y.P. Sheng, 1982; "ADI Procedures for Solving the Shallow-Water Equations in Transformed Coordinates," Proc. 1982 Army Numerical Analysis and Computer Conference.
- Cheng, R.T., T.M. Powell, and T.M. Dillon, 1976; "Numerical Models of Wind-Driven Circulation in Lakes," Appl. Math Modeling, 1, pp. 141-159.
- Davis, R.E., R. de Szoeke, D. Halpern, and P. Miller, 1981a; "Variability in the Upper Ocean During MILE, 1, the Heat and Momentum Balances," Deep Sea Res., in press.
- Donaldson, C.duP., 1973; "Atmospheric Turbulence and the Dispersal of Atmospheric Pollutants," in AMS Workshop on Micrometeorology (U.A. Haugen, ed.), Science Press, Boston, pp. 313-390.
- Edinger, J.E. and E.M. Buchak, 1979; "Preliminary LARM Simulation of the WES GRH Flume," Report to Waterways Experiment Station.
- Eriksen, C.C., 1978; "Measurements and Models of Fine Structure, Internal Gravity Waves and Wave Breaking in the Deep Ocean," J. Geophys. Res., 83, 2989-3009.

- Forristal, G.Z., R.C. Hamilton, and V.J. Cardone, 1977; "Continental Shelf Currents in Tropical Storm Delia: Observations and Theory," J. Phys. Oceano., 7, pp. 532-546.
- Gedney, R.T. and W. Lick, 1972; "Wind-Driven Current in Lake Erie," J. Geophys. Res., 77, No. 15.
- Grant, W., 1981; Personal Communication.
- Haq, A., W. Lick, and Y.P. Sheng, 1974; "The Time-Dependent Flow in Large Lakes with Applications to Lake Erie," Technical Report, Dept. Earth Sciences and Dept. Mechanical and Aerospace Engineering, Case Western Reserve University.
- Ippen, A.T. and D.R.F. Harleman, 1961; "Analytical Studies of Salinity Intrusion in Estuaries and Canals," Phase 1: One-Dimensional Analysis, Technical Bulletin No. 5, Committee on Tidal Hydraulics, U.S. Army Corps of Engineers.
- Leendertse, J.J., 1970; "A Water Quality Simulation Model for Well-Mixed Estuaries and Coastal Seas, I: Principles of Computation," RM-6230-rc, Rand Corporation, Santa Monica, CA.
- Leendertse, J.J. and S.K. Liu, 1975; "A Three-Dimensional Model for Estuaries and Coastal Seas, II: Aspects of Computation," Rand Report R-1764-OWRT.
- Lewellen, W.S. and Y.P. Sheng, 1980; "Modeling of Dry Deposition of SO_2 and Sulfate Aerosols," Report EPRI EA-1452, Electric Power Research Institute, Palo Alto, CA.
- Munk, W.H. and E.P. Anderson, 1948; "Notes on the Theory of the Thermocline," J. Mar. Res., 1, 276-295.
- Orszag, S.A. and M. Israeli, 1974; "Numerical Simulation of Viscous Incompressible Flows," Annual Review of Fluid Mechanics, 6, 281-318.
- Sheng, Y.P., 1975; "Wind-Driven Currents and Dispersion of Contaminants in the Near-Shore Regions of Large Lakes," Contract Report H-75-1, Waterways Experiment Station.
- Sheng, Y.P., W. Lick, R. Gedney, and F. Molls, 1978; "Numerical Computation of the Three-Dimensional Circulation in Lake Erie; A Comparison of a Free-Surface and a Rigid-Lid Model," J. Phys. Oceano., 8, pp. 713-727.
- Sheng, Y.P., 1980; "Modeling Sediment Transport in a Shallow Lake," in Estuaries and Wetland Processes (P. Hamilton, ed.), Springer-Verlag, Berlin, Heidelberg, pp. 299-337.

Sheng, Y.P. and W.S. Lewellen, 1982; "Current and Wave Interaction Within the Benthic Boundary Layer," EOS, 63, 3, pp. 72-73.

Sheng, Y.P. and W. Lick, 1972; "Wind-Driven Currents in a Partially Ice-Covered Lake," Proc. 16th Conf. Great Lakes Research, 1001-1008.

Sheng, Y.P. and W. Lick, 1980; "A Two-Mode Free-Surface Numerical Model for the Three-Dimensional Time-Dependent Currents in Large Lakes," EPA Report 600/3-80-047, 62 pp.

Sheng, Y.P., H. Segur, and W.S. Lewellen, 1978; "Applications of a Spatial Smoothing Scheme to Control Short-Wave Numerical Oscillations," A.R.A.P. Tech. Memo No. 78-8, 14 pp.

Sternberg, R.W., 1972; "Predicting Initial Motion and Bedload Transport of Sediment Particles in the Shallow Marine Environment," in Shelf-Sediment Transport, (D.P. Swift, ed.), Dowden, Hutchinson, Ross, Stroudsburg, PA.

van de Kreeke, J. and S.S. Chiu, 1980; "Tide-Induced Residual Flow," in Mathematical Modeling of Estuarine Physics, (J. Sunderman, ed.), Springer-Verlag, Berlin, Heidelberg, New York.

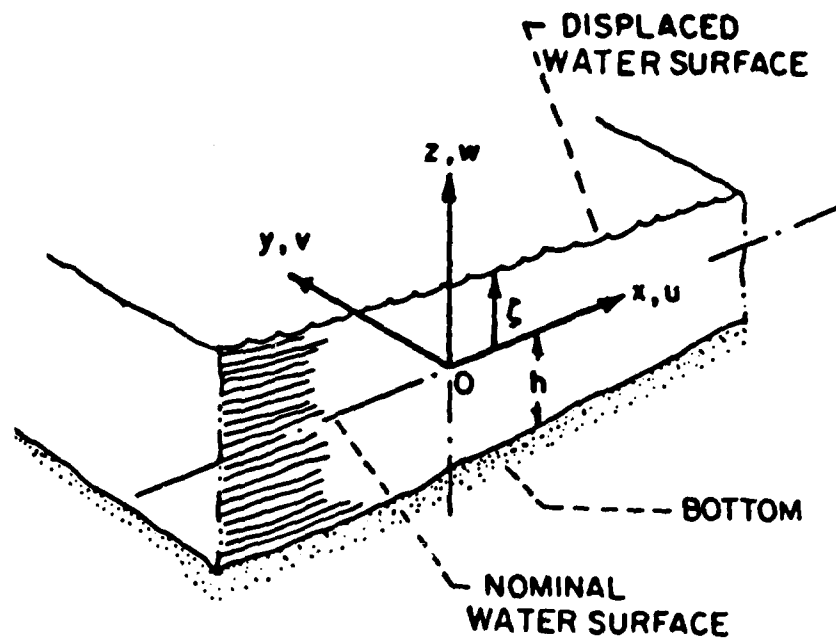


Figure 1. Cartesian coordinates located at the nominal water surface.

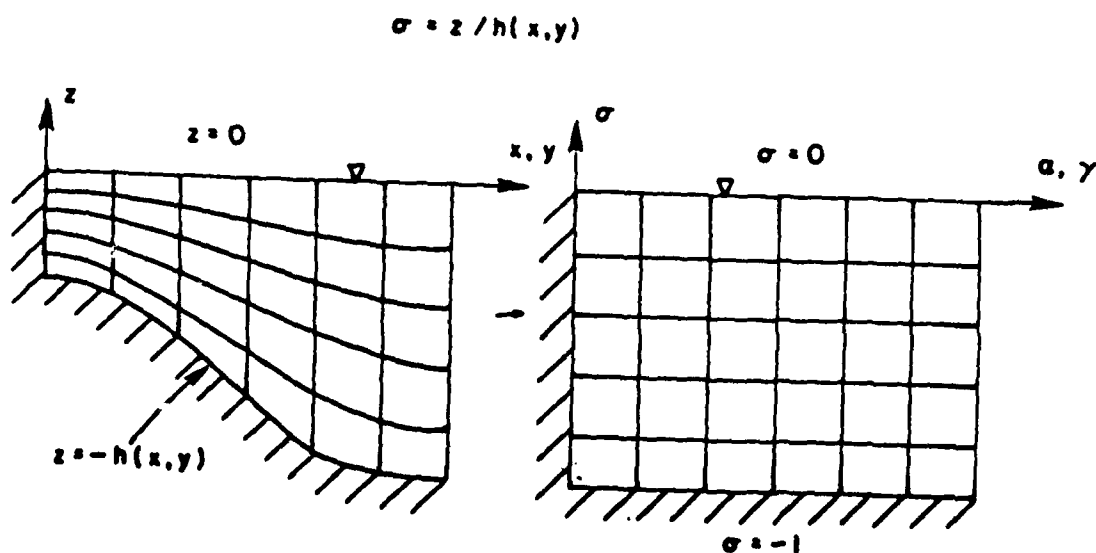


Figure 2. Vertical stretching of the coordinates.

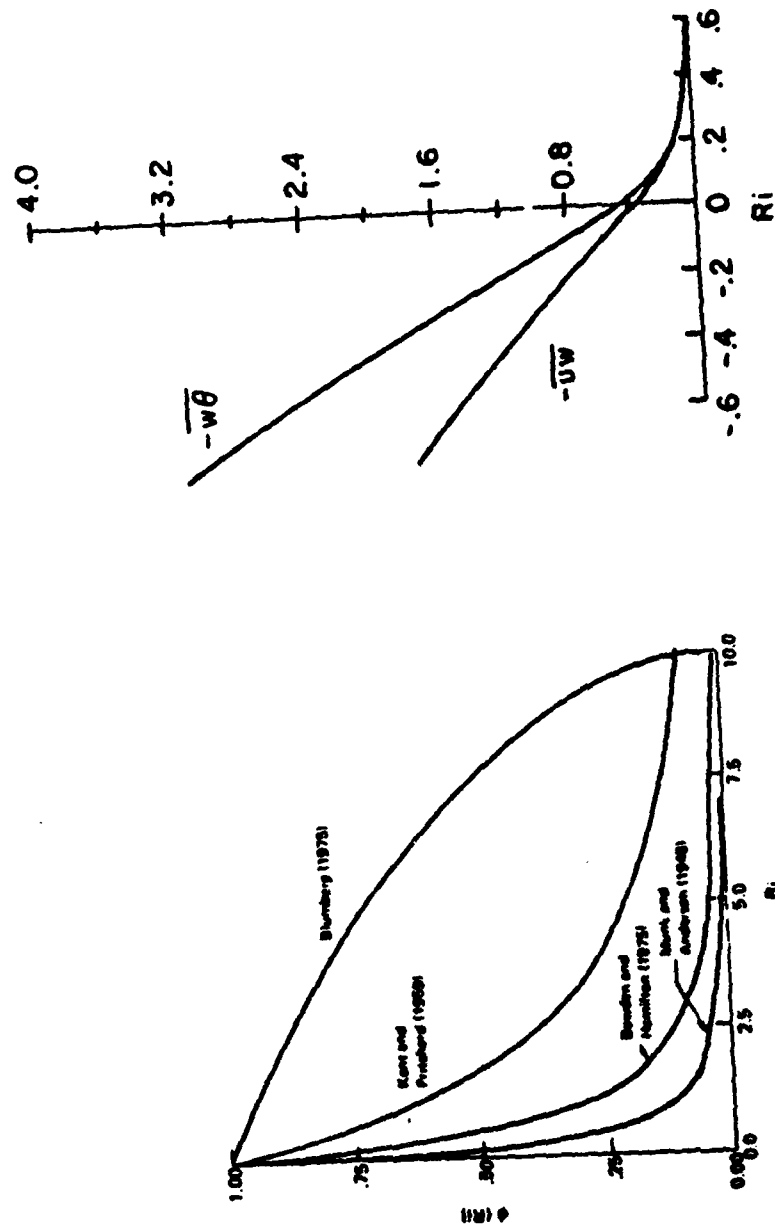


Figure 3. (a) Various empirical formulations of the stability function $\phi(Ri)$.
 (b) "Superequilibrium" Reynolds stress profile and heat flux profile normalized
 by $\Lambda^2 (du/dz)^2$ and $\Lambda^2 (d\theta/dz)$, respectively.

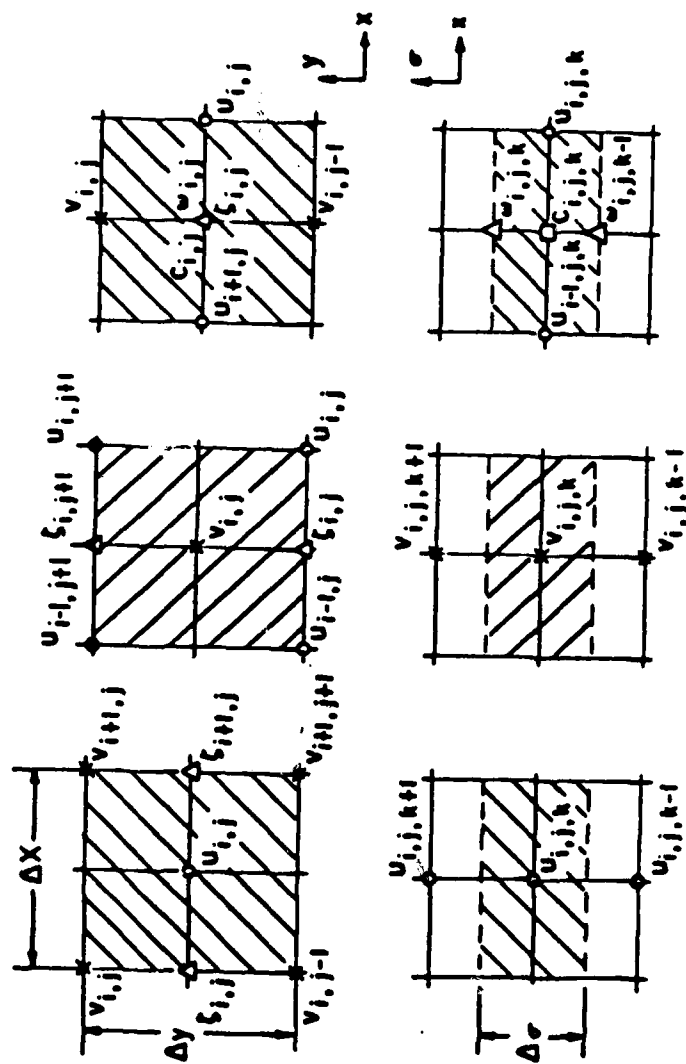


Figure 4. Basic numerical grid structure in the vertically and horizontally stretched coordinates.

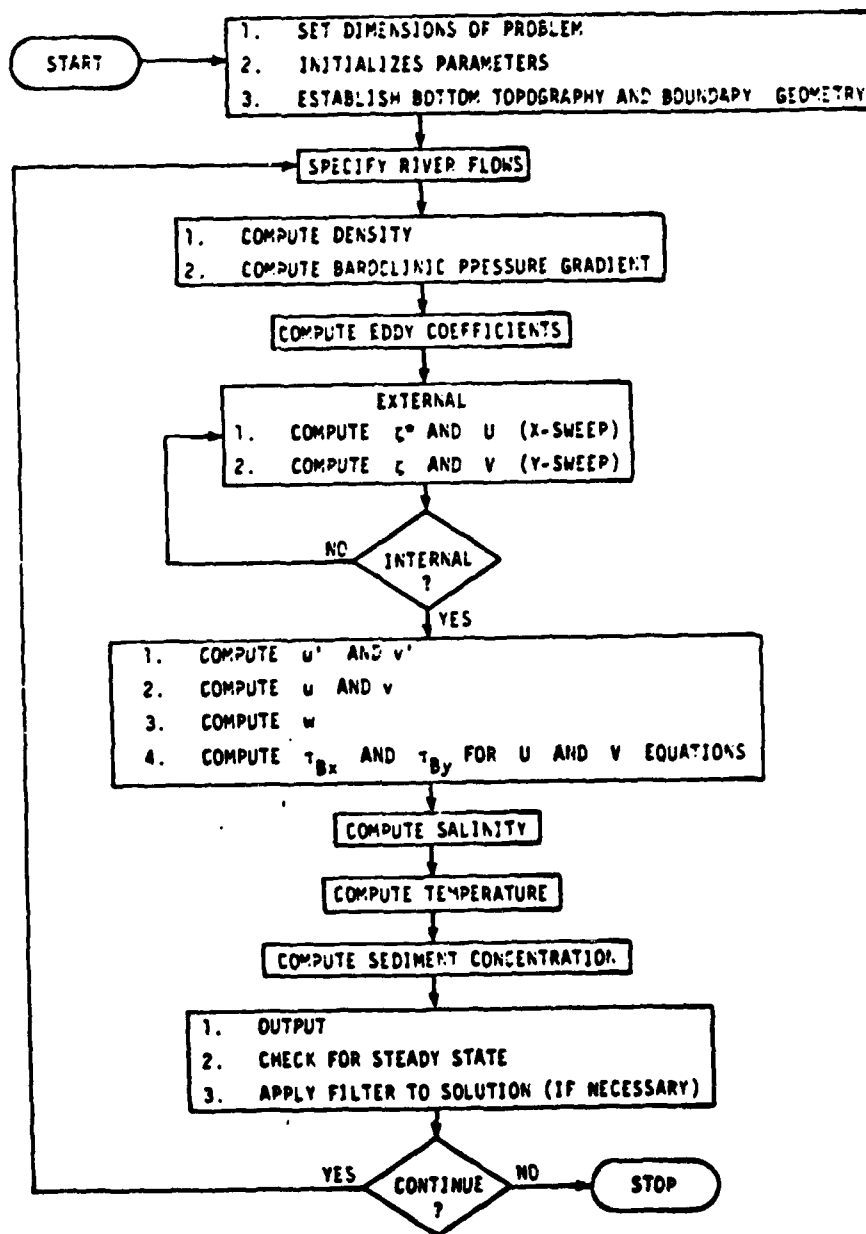


Figure 5. Flow chart of the solution algorithm of the three-dimensional hydrodynamic model.

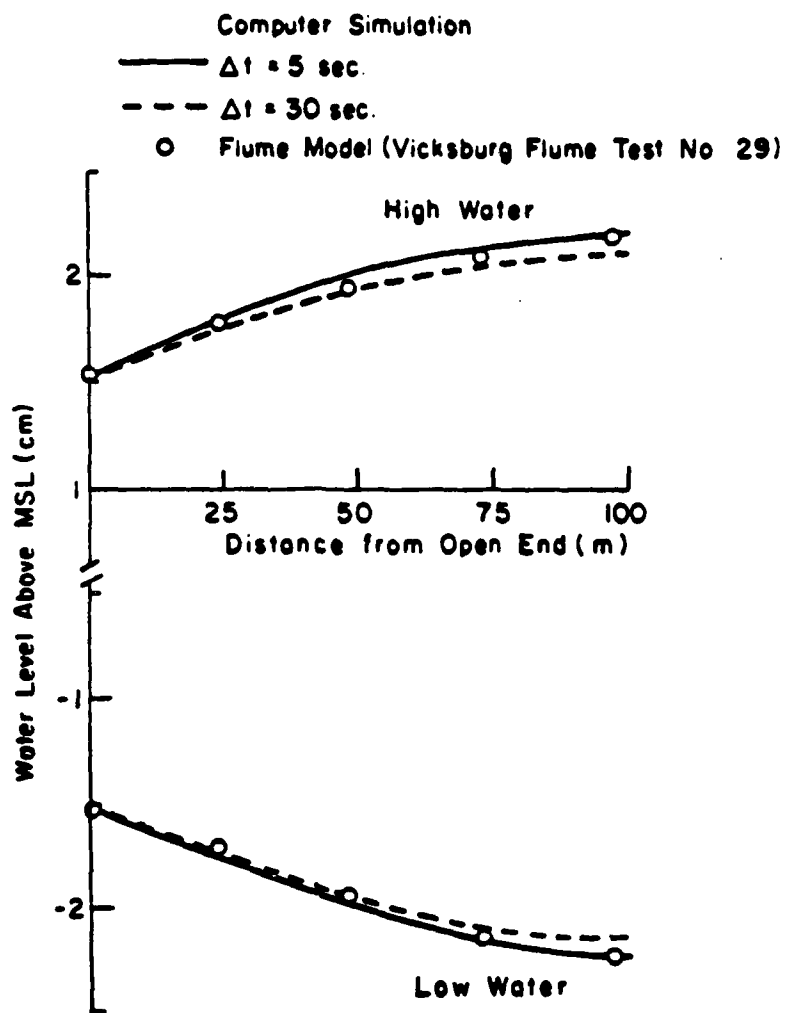


Figure 6. Computer simulation of the Vicksburg Tidal Flume.

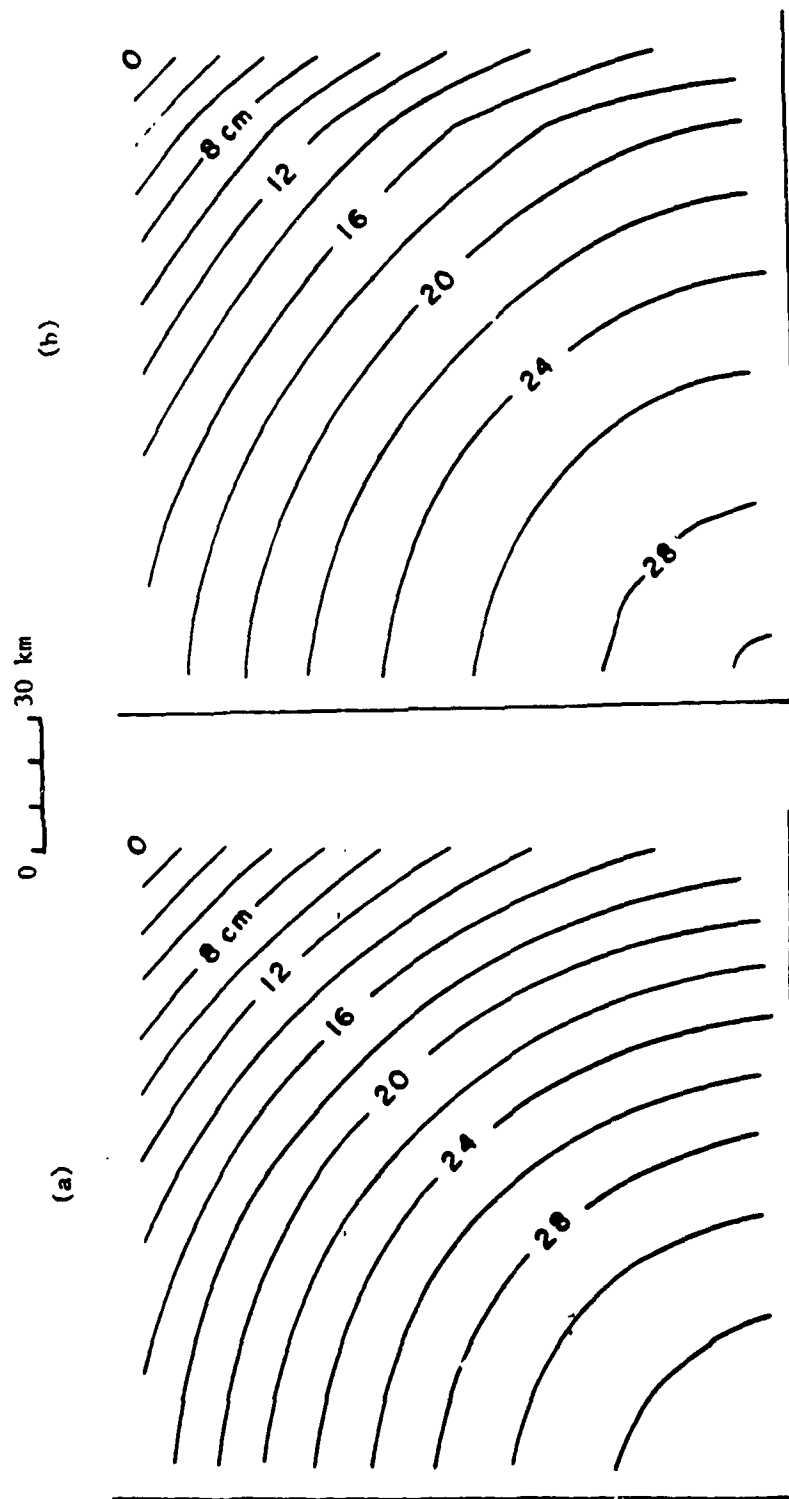


Figure 7 (a) and (b). Water level at the peak of ebb tide in an open bight driven by tidal waves along the open boundaries: (a) analytical result; (b) numerical result.

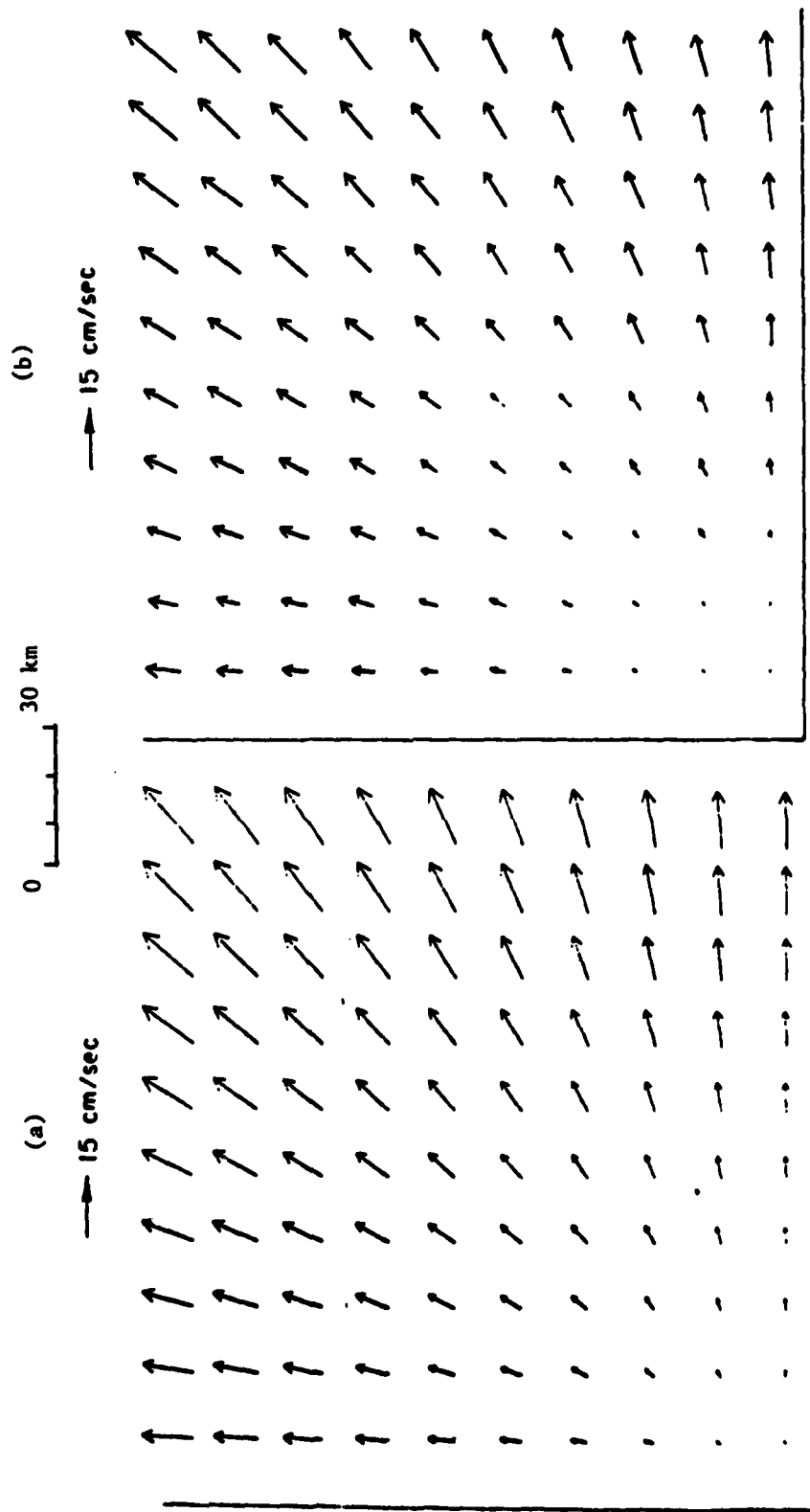


Figure 8 (a) and (b). Vertically-averaged velocities at the peak of ebb tide in an open bight driven by tidal waves along the open boundaries: (a) analytical result; (b) numerical result.

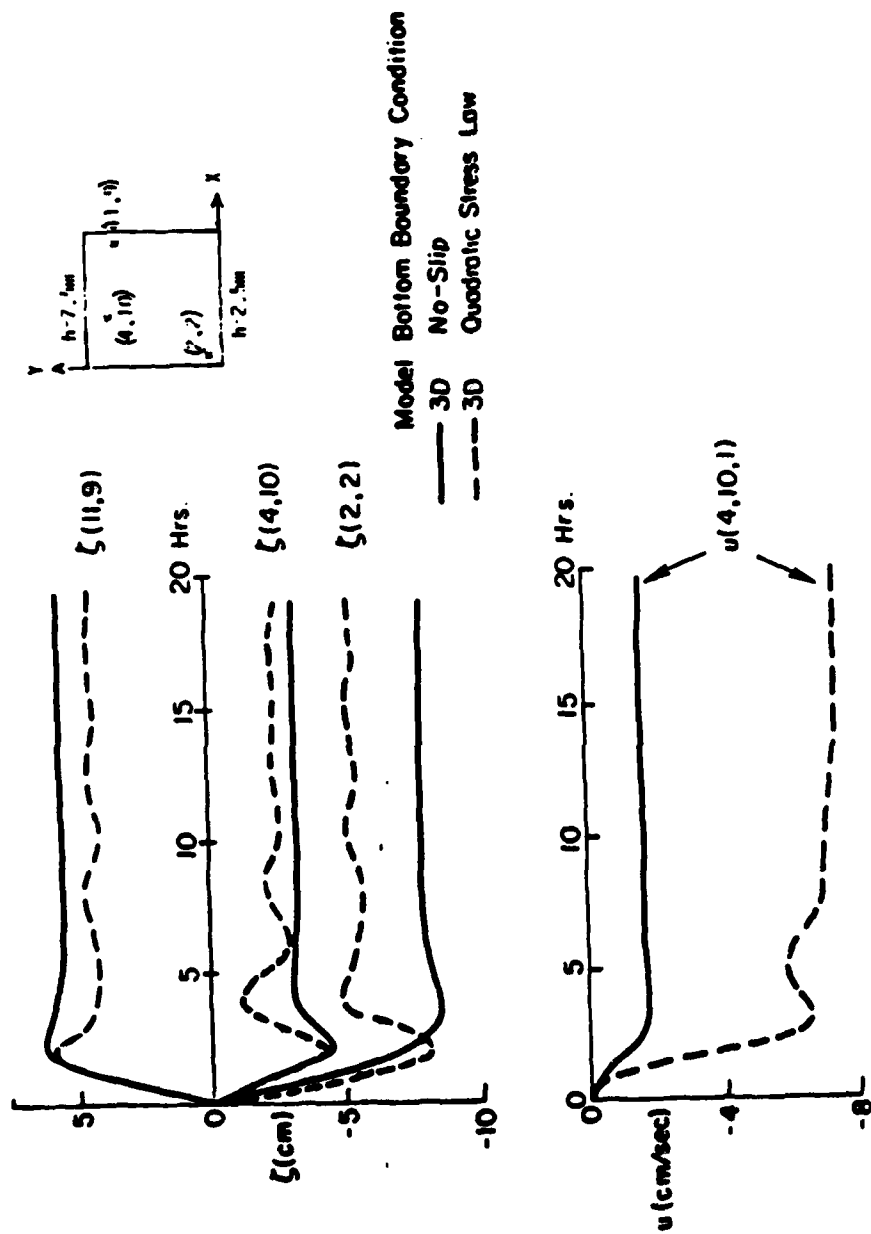


Figure 9. Water level and horizontal velocity at selected points in a constant slope enclosed basin driven by a uniform wind stress.

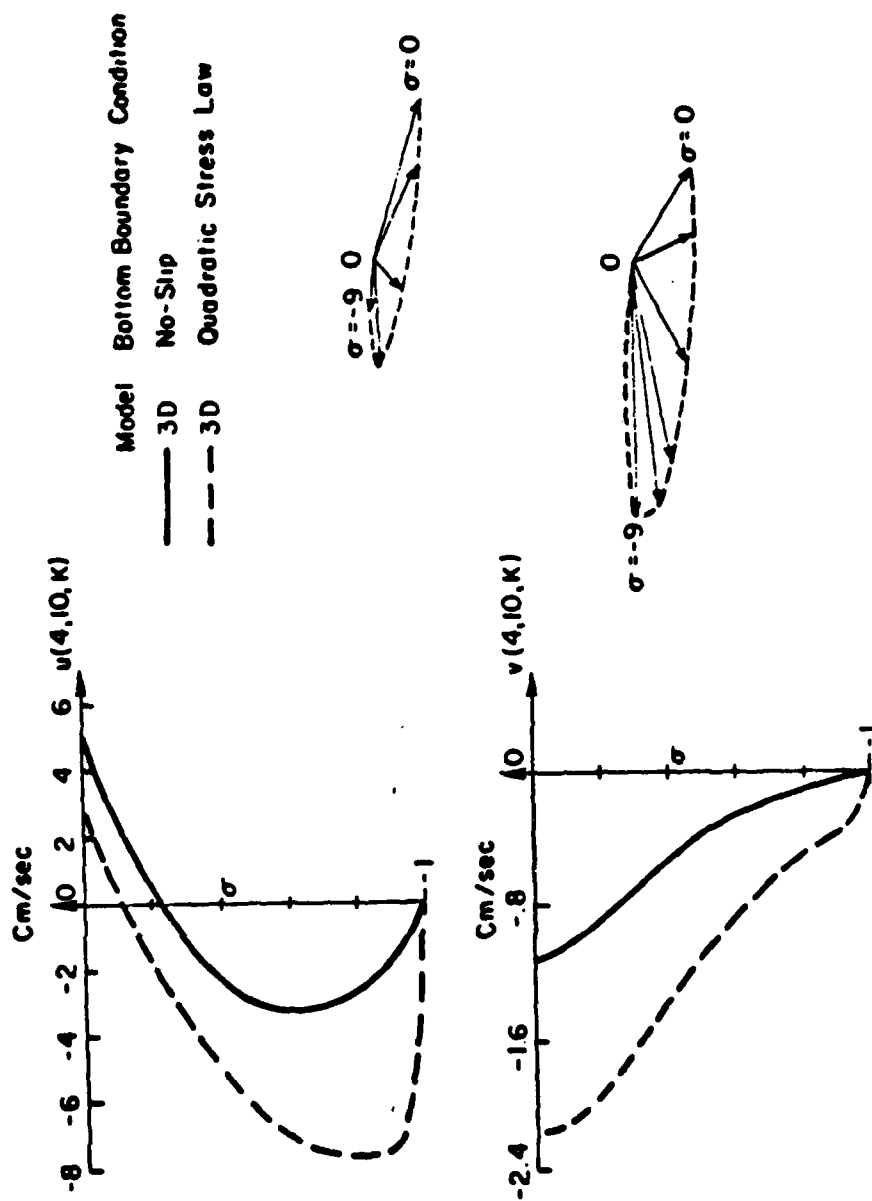


Figure 10. Vertical profile of steady-state horizontal velocities at a point in the enclosed basin.

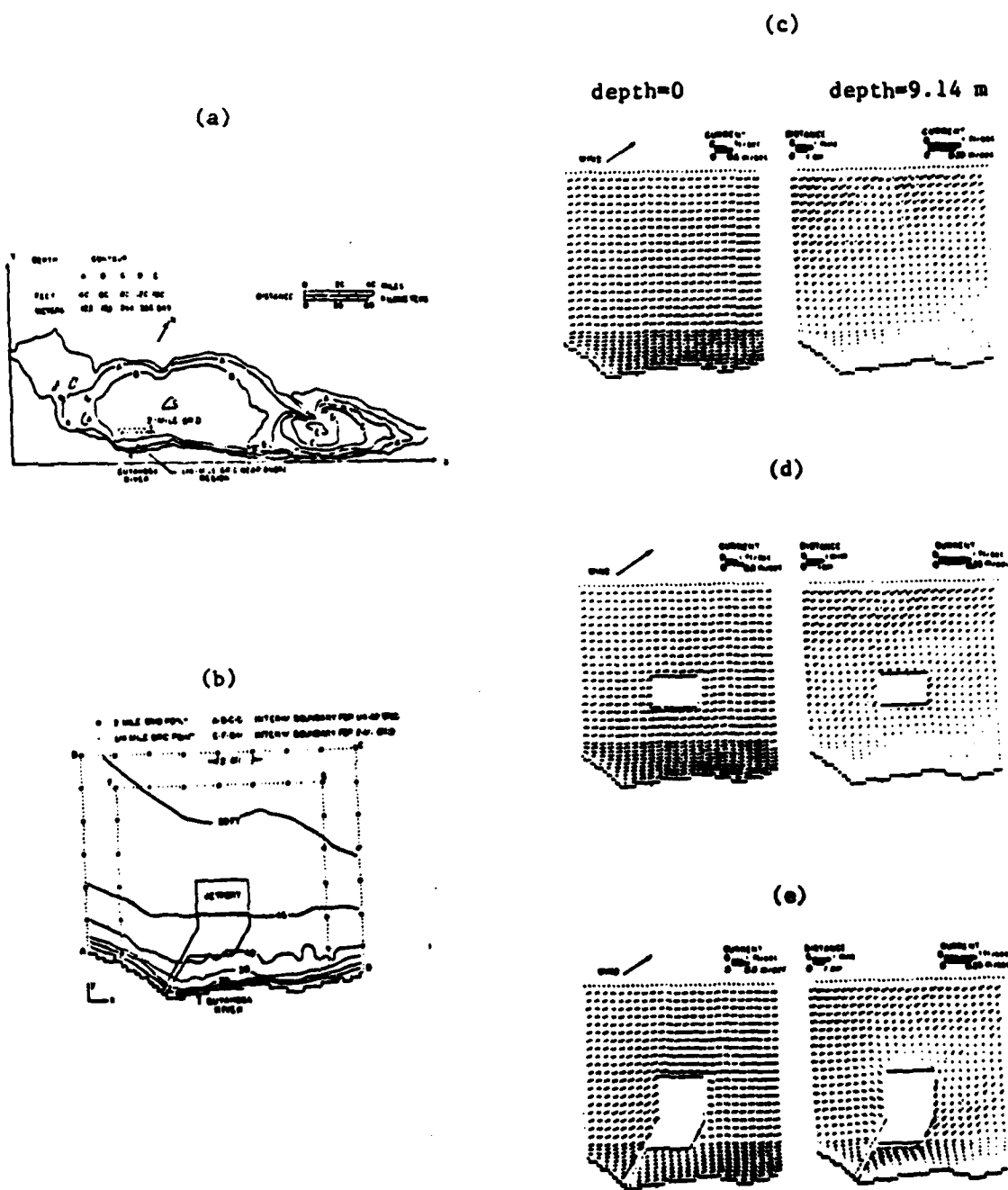


Figure 11 (a), (b), (c), (d) and (e). Steady-state horizontal velocities in the near-shore Cleveland area of Lake Erie caused by a 7.6 m/sec wind from SSW: (a) bottom topography of Lake Erie; (b) bottom topography and grid structure in the near-shore; (c) horizontal velocities in the near-shore at the water surface and 9.14 m depth; (d) horizontal velocities in the presence of a jetport island; and (e) horizontal velocities in the presence of a jetport island with a causeway.

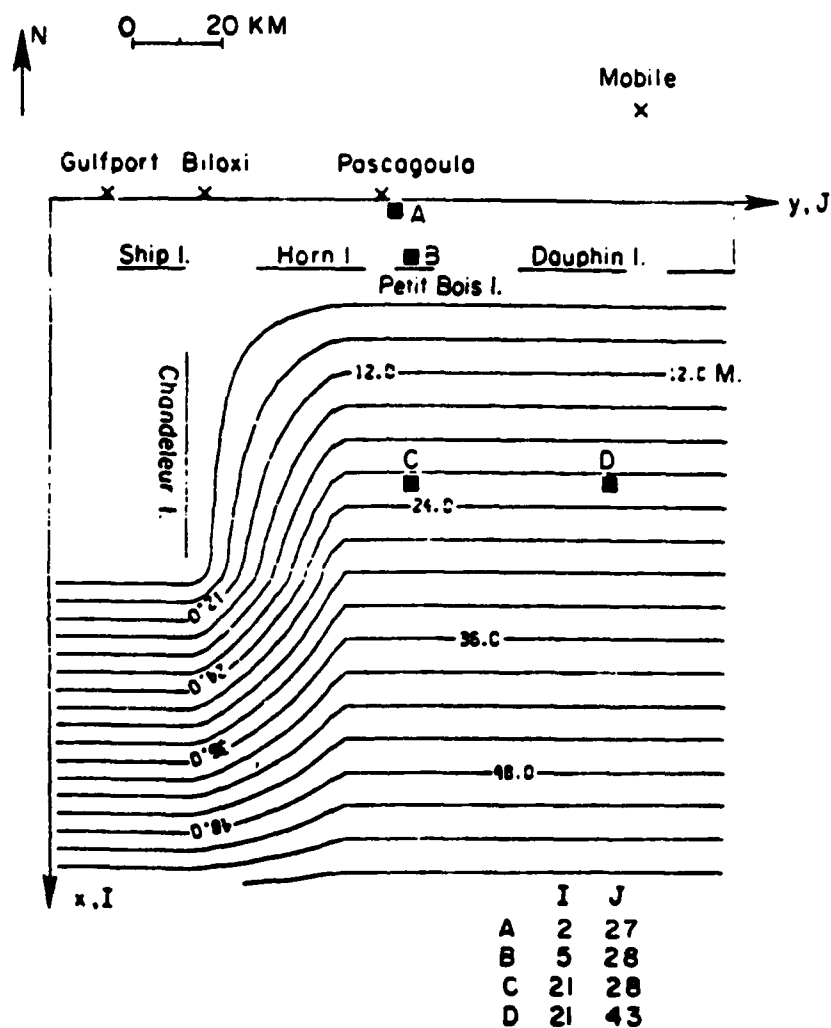


Figure 12. Simplified geometry and topography for the Mississippi Sound and adjacent continental shelf waters.

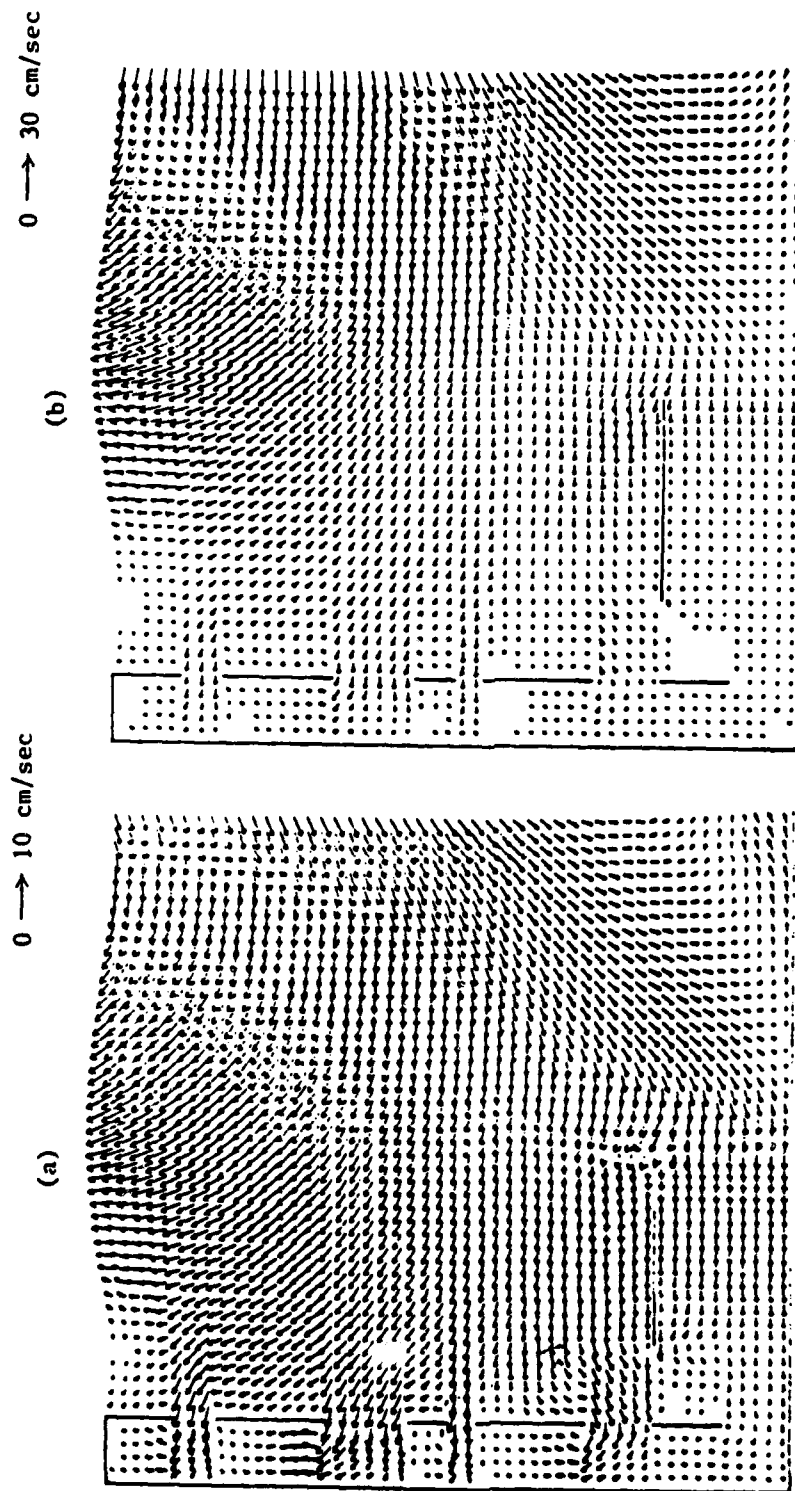


Figure 13 (a) and (b). Tide-induced horizontal velocities in the Mississippi Sound and adjacent shelf waters at the end of a four-day simulation: (a) at $\sigma=-0.1$; (b) at $\sigma=-0.9$.

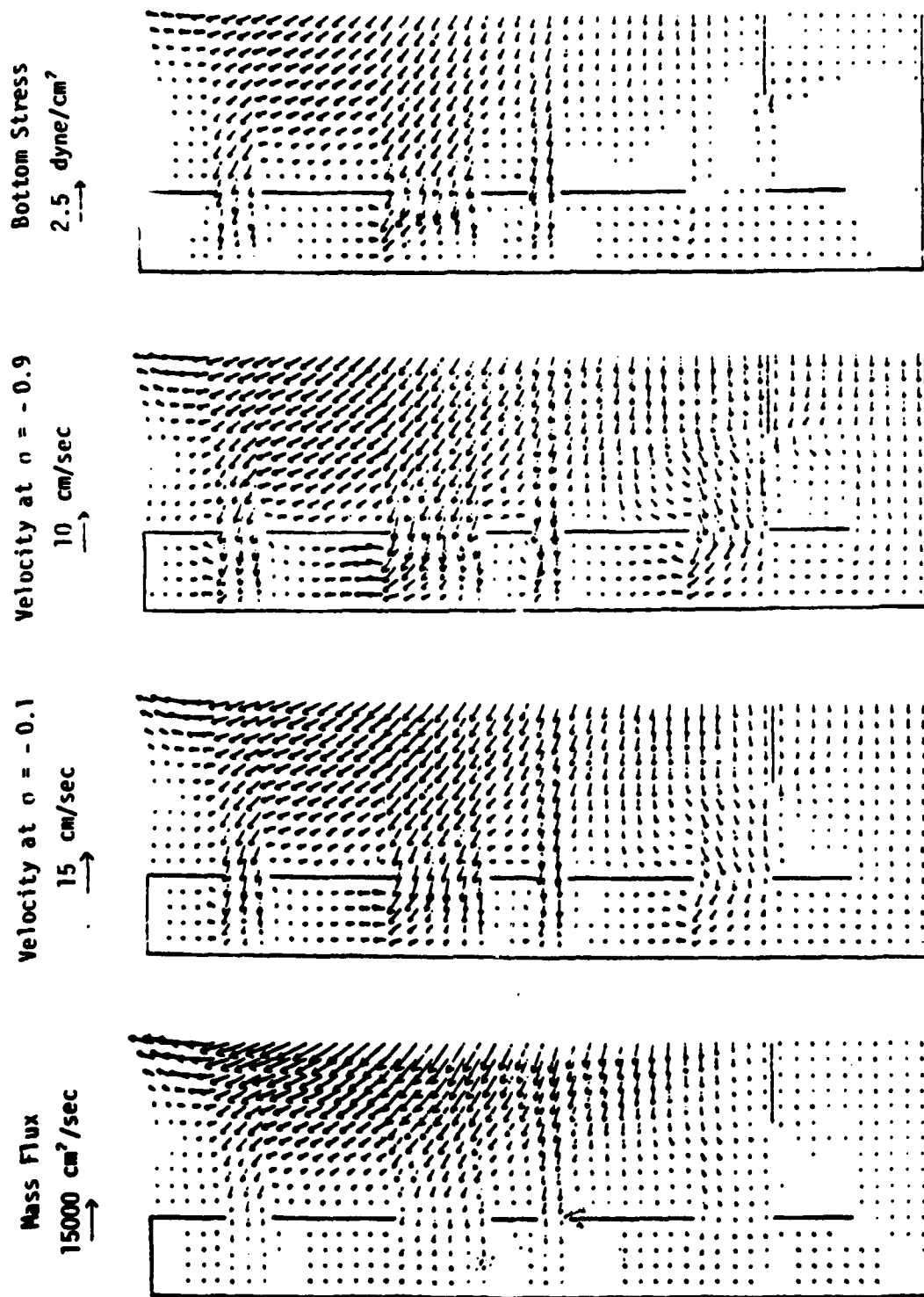


Figure 14. Tidal currents and bottom shear stress within the Mississippi Sound and nearby waters after a four day simulation: (a) vertically-integrated velocities, (b) velocities at $\sigma = -0.1$, (c) velocities at $\sigma = -0.9$, and (d) bottom shear stress.

Bottom Stress
1.2 dyne/cm²
→

Velocity at $\sigma = -0.9$
20 cm/sec
→

Velocity at $\sigma = -0.1$
30 cm/sec
→

Mass Flux
15000 cm²/sec
→

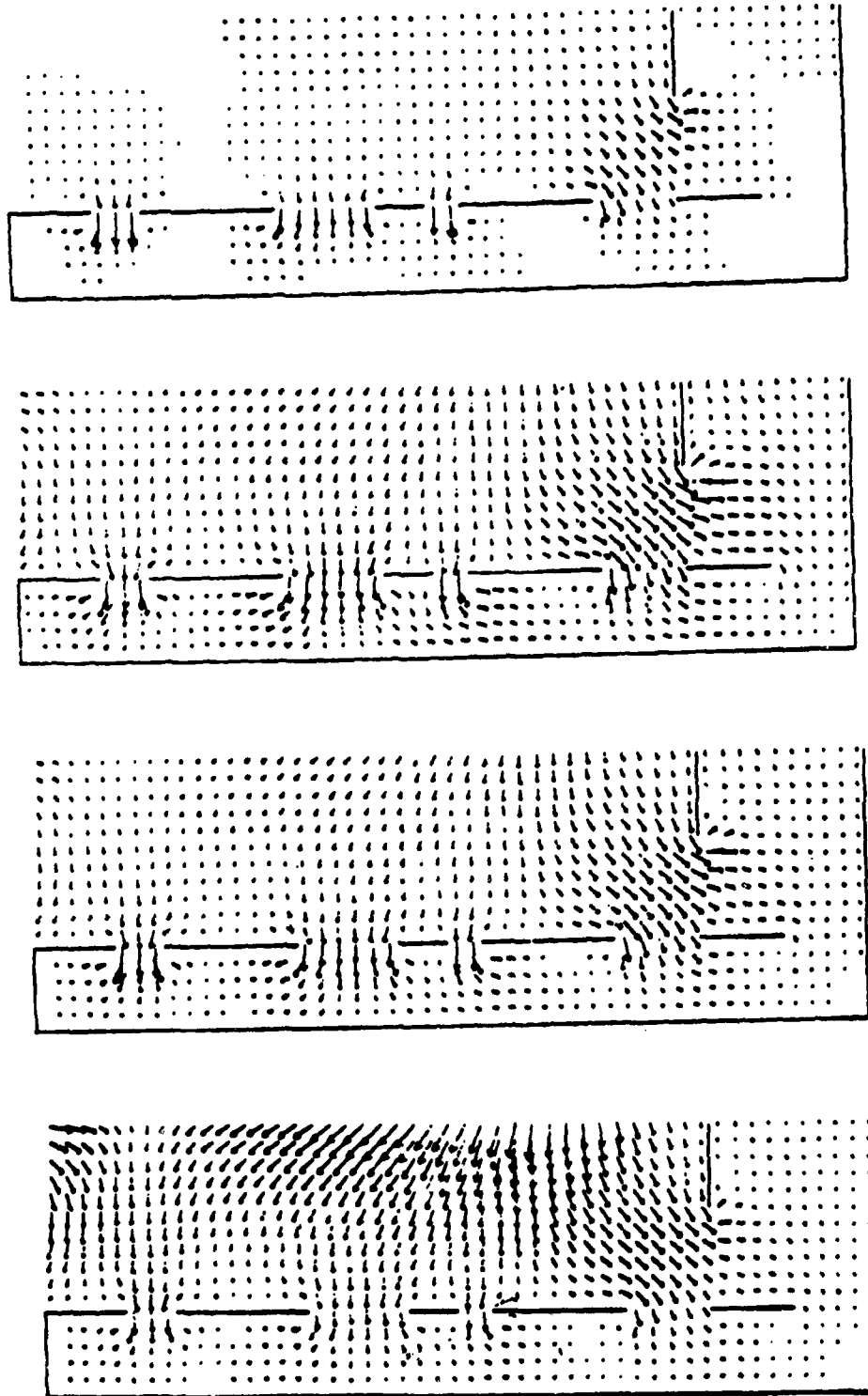


Figure 15. Same as Figure 14, except at six hours later.

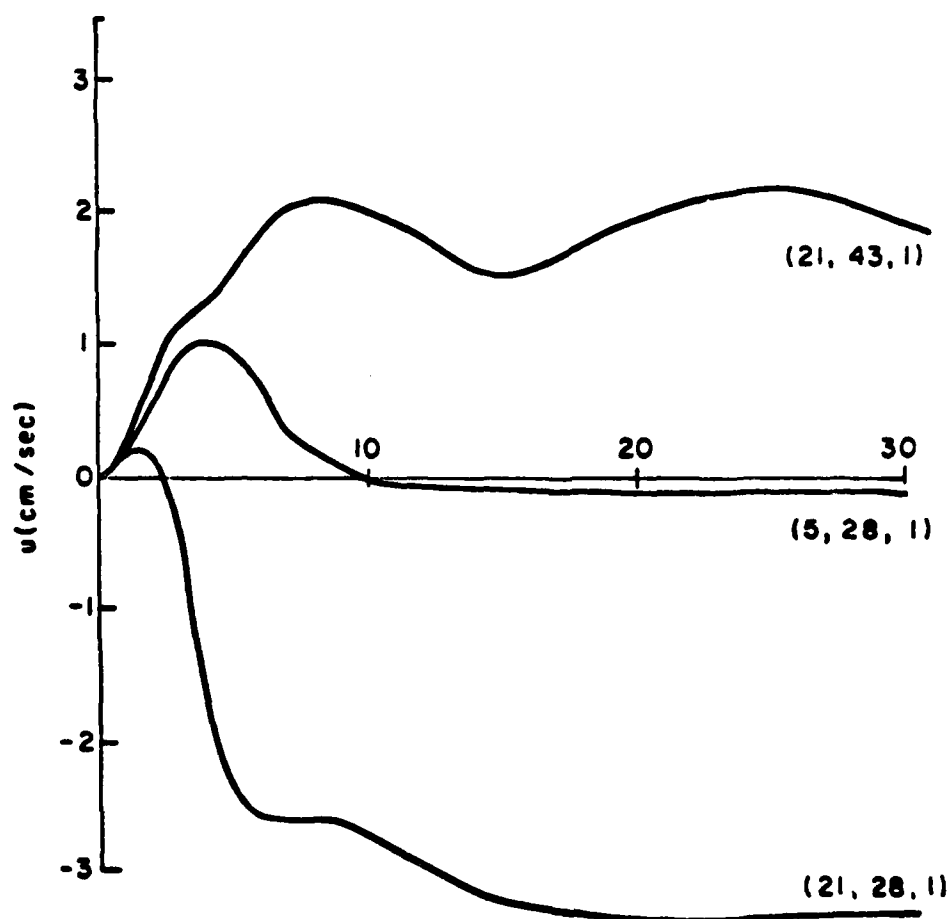


Figure 16. Horizontal velocities near the bottom ($\sigma=-0.9$) as a function of time for three locations.

SOLUTION ADAPTIVE GRIDS FOR PARTIAL DIFFERENTIAL EQUATIONS

Dale A. Anderson
Department of Aerospace Engineering and
Computational Fluid Dynamics Institute
Iowa State University
Ames, Iowa

ABSTRACT. Various techniques for constructing solution adaptive grids used in numerically solving partial differential equations are reviewed. These methods include those which directly determine the metrics or coordinates and schemes which postulate laws of point motion providing a direct calculation of the grid speed. Examples showing results obtained with each method are presented and suggestions for possible directions of future work are made.

1. INTRODUCTION. The selection of an appropriate coordinate system and grid is an important consideration in the numerical solution of partial differential equations. In most problems, the physical domain is transformed into a computational domain and the numerical solution is obtained in computational space. For simplicity the computational domain is usually rectangular and the mesh points are uniformly distributed. The physical domain boundaries are selected to simplify boundary condition application or provide other advantages in the computation. The transformation which maps the physical domain into computational space is the subject of grid generation.

Numerous methods for generating appropriate grids have been proposed. These methods can be roughly classified into differential equation techniques, algebraic schemes, and classical complex variable methods. Each of these provides special properties which may be used to eliminate certain problems. For example a simple compression mapping can be used to cluster points in a controlled region near a boundary. This will provide adequate resolution in regions where rapid changes of the dependent variables occur. A typical example where this type of transformation is used is in resolving the boundary-layer region in a fluid mechanics problem.

In computing the numerical solution of a partial differential equation, the first task is that of generating an acceptable grid. Once the grid is established, the distribution of points in physical space never changes unless the grid is restructured during the calculation. The disadvantage in maintaining a fixed grid is that an a priori knowledge of the solution is required. As the solution evolves, the grid should change to provide adequate mesh point density in physical space where it is needed. Ideally, the grid should be adaptive. This requires that the grid evolve as part of the solution to the problem. If an adaptive grid is used, a numerical solution of a partial differential equation must be computed, and in addition, the mapping relating the physical and computational domains must be determined.

A number of techniques for generating solution adaptive grids used in conjunction with finite-difference methods are reviewed in the following sections. These techniques are generally of the differential equation type although some may be classed as hybrid differential-algebraic schemes. Attention is focused on those

AD-A118 920

ARMY RESEARCH OFFICE RESEARCH TRIANGLE PARK NC
PROCEEDINGS OF THE 1982 ARMY NUMERICAL ANALYSIS AND COMPUTERS C--ETC(U)
AUG 82
ARO-82-3

F/O 12/1

UNCLASSIFIED

NL

7 7

LIBRARY



END

DATE

FILMED

10 82

DTIC

which redistribute a fixed number of grid points to achieve an improved solution when compared to that obtained on a nonadaptive grid.

2. METHOD CLASSIFICATION. Techniques for constructing solution adaptive grids can be classified according to the approach selected in constructing the necessary mapping. An example showing the transformation of the first-order linear wave equation from physical to computational space will demonstrate this point. Consider the equation

$$(1) \quad \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

where (x,t) are the independent variables, u is the dependent variable, and c is the constant wave speed. Suppose a real, nonsingular mapping to computational space of the form

$$(2) \quad \begin{aligned} \tau &= t \\ \xi &= \xi(x,t) \end{aligned}$$

is used where (ξ, τ) are the computational coordinates. Equation (1) in computational space becomes

$$(3) \quad \frac{\partial u}{\partial \tau} + (\xi_t + c \xi_x) \frac{\partial u}{\partial \xi} = 0$$

This partial differential equation is integrated with respect to τ to obtain a solution for $u(x,t)$. The central problem which must be solved to generate an adaptive grid is to determine the transformation $\xi = \xi(x,t)$ as needed in the integration of Eq. (3).

The simple form of Eq. (3) suggests a way of classifying methods in evaluating $\xi = \xi(x,t)$. Two terms defining the relation between the physical and computational domains must be evaluated. The first, ξ_t , is termed the grid speed. While grid points in computational space do not move, this is an appropriate name since the grid speed at any point in physical space is given by

$$(4) \quad x_\tau = - \frac{\xi_t}{\xi_x}$$

The metric of the transformation, ξ_x , also appears explicitly in Eq. (3). For transformations of the type given by Eq. (2) we may write

$$(5) \quad x_\xi = \frac{1}{\xi_x}$$

Two methods of constructing the transformation, $\xi(x,t)$, can now be identified. The first is to evaluate the metric, ξ_x , using some rule and then determine the new locations of points in the physical plane using this knowledge. Once the new x locations are known, the grid speed, x_τ or ξ_t , can be evaluated using the time history of the grid point motion. Techniques using this approach will be referred

to as type A schemes. The second method, which we shall refer to as a type B scheme, directly provides the grid speed. This is obtained by postulating a law or set of laws that control grid motion. The grid speed equation is integrated giving the new mesh point locations at the same time the governing partial differential equation is integrated. Once the new grid is known in physical space, the metric is calculated.

There are advantages and disadvantages to both approaches. Type A schemes are physically more obvious. Essentially a new grid can be constructed at any time during the calculations. On the other hand, the dynamic coupling of the grid with the partial differential equation through ξ_i is usually lagging in time even though points are precisely positioned as desired. Extension of these schemes to multidimensional problems is also difficult. Type B schemes are easily used in multidimensional problems and the dynamic coupling is correct in time. However, grid point control is difficult and proper laws governing grid speed must be carefully formulated. Various methods of both types are reviewed in the following sections, and typical grids produced using these schemes are presented.

3. TYPE A METHODS. The type A schemes can be viewed as regrid procedures which are employed after each integration step or at the end of any predetermined number of steps. A number of methods of this type having different degrees of complexity have been developed by various investigators.

Dwyer et al. [5] developed an adaptive grid procedure for use in solving both time-dependent and steady problems in fluid dynamics and heat transfer. This scheme is designed to provide adequate resolution in high gradient regions. To demonstrate the application of Dwyer's method, consider the physical domain and the corresponding computational plane shown in Figure 1. Suppose we wish to resolve regions of high gradient by placing more points in those regions and fewer points in regions of low gradient. If the dependent variable requiring better resolution is the temperature, T , in a heat conduction problem, a point clustering in high gradient regions can be achieved if the relationship between physical and computational space is given by

$$(6) \quad d\xi \propto \left| \frac{\partial T}{\partial s} \right| ds$$

where s represents arc length along the $\eta = \text{constant}$ lines in the physical domain.

If the maximum value of ξ is 1, Eq. (6) may be written in the form

$$(7) \quad \xi(x,y,t) = \frac{\int_0^s (1 + b \left| \frac{\partial T}{\partial s} \right|) ds}{\int_0^{s_{\max}} (1 + b \left| \frac{\partial T}{\partial s} \right|) ds}$$

where s_{\max} is the largest value of s encountered in physical space, and b is a constant or a function which permits the gradient sensitivity of the transformation to be adjusted. If b is zero, the mapping defined by Eq. (7) provides a uniform distribution of points given by

$$(8) \quad \xi(x, y, t) = \frac{\int_0^s |dT|}{\int_0^{s_{\max}} |dT|}$$

This provides a grid where the point locations in physical space are at surfaces of constant difference in the dependent variables. Dwyer notes that this procedure can lead to difficulties in zero gradient regions where large values of the second derivative term exist. This problem can be avoided by adding a second derivative term to the transformation defined by Eq. (7).

Physical coordinates are recovered by replacing the integrals in Eq. (7) by simple quadratures. For a given value of ξ , the correct value of s is obtained by integrating and interpolating. Once the physical coordinates are computed, the metrics of the transformation can be obtained using finite differences and the grid speed is determined by using a backward difference.

Figure 2 presents results for a two-dimensional transient heat conduction problem at an intermediate time. In this problem, the initial temperature was set equal to zero everywhere. The temperature was impulsively raised to a constant value along the lower boundary. As time progresses, heat flows into the domain from this boundary creating large temperature gradients there. The grid structure shows the clustering of points in this high gradient region and the temperature distribution shows that isotherms correspond to the grid structure as desired.

Klopfer and McRae [8] developed a method of adjusting mesh point locations while calculating the flow in a shock tube. The mesh was adjusted in an attempt to reduce the error in the numerical solution for the flow. The governing conservation equations for flow in a shock tube written in computational coordinates (ξ, τ) are

$$(9) \quad (x_{\xi} \vec{w})_{\tau} + (\vec{f} - x_{\tau} \vec{w})_{\xi} = 0$$

where

$$(10) \quad \vec{w} = (\rho, \rho u, e)^T, \quad \vec{f} = [\rho u, p + \rho u^2, (e + p)u]^T$$

In these expressions, ρ is the density, p is the pressure, u is the velocity, and e is the energy. If a second-order finite-difference scheme is applied to Eq. (9) and the resulting discretized equation is expanded, the modified partial differential equation is obtained. If the first truncation error term is retained, this expression is of the form

$$(11) \quad (x_{\xi} \vec{w})_{\tau} + \left[\vec{f} - x_{\tau} \vec{w} + \frac{\Delta \xi^2}{6} (\vec{f} - x_{\tau} \vec{w})_{\xi \xi} \right]_{\xi} = 0$$

The flux quantity is altered by the error term given by

$$(12) \quad \vec{R} = \frac{\Delta \xi^2}{6} (\vec{f} - x_{\tau} \vec{w})_{\xi \xi}$$

This error term can be reduced by changing the mesh point locations if the point motion is driven by the error. This error term has three scalar components due to each of the scalar conservation equations: continuity, momentum, and energy. Define a scalar cluster function, $r(\xi)$, composed of a weighted sum of the scalar components of \bar{R} .

$$(13) \quad r(\xi) = S[Ar_1 + Br_2 + Cr_3]$$

In this expression S is a smoothing operator and A , B , and C are functions which can be adjusted to provide different weighting to the respective error components. The adaptive grid is constructed by assuming that the metric, x_ξ , is proportional to $r(\xi)$. In particular Klopfer and McRae used the relation

$$(14) \quad x_\xi = (x_\xi)_{ave} \left[(1 - r(\xi)/r_{max})(K_{max} - K_{min}) + K_{min} \right]$$

where

$$(x_\xi)_{ave} = \text{average on the mesh}$$

$$r_{max} = \max[r(\xi)]$$

and K_{max} and K_{min} are constants which control the amount of clustering. Once the metrics are defined, the physical coordinates are recovered by integrating Eq. (14), and the grid speed, x_t , is obtained by using a backward difference.

Figure 3 shows a comparison of results for a shock tube for both a fixed and an adaptive mesh. In both cases smoothing was added to reduce oscillations, and MacCormack's second-order method was used to calculate the solution. The results show the smooth expansion wave moving to the left into the high pressure region, and the shock wave moves into the low pressure side with the contact surface separating the fluid originally on each side of the diaphragm at $t = 0$. The clustered results show no oscillations at the shock or contact surface demonstrating the effectiveness of the clustering technique. It should be noted that as many as five or six points are used to define the shock and contact surface so the discontinuities do not appear between two mesh points.

The two methods reviewed should be referred to as error reducing schemes. No attempt has been made to show that adaptive grids produced are minimum error grids. If formal minimization techniques are used to produce the grid, the grid would be a minimum error solution consistent with the error measure used. Several authors have used this approach. Yanenko et al. [15] constructed an adaptive grid by minimizing a linear combination of variables related to mesh quality. The first quantity was mesh distortion and is a measure of the non-orthogonality of the mesh. The second term provided an estimate of how well the mesh moved with the fluid and the third provided a reduction in mesh spacing in high gradient regions. Ablow and Schecter [1] solved the two-dimensional Poisson equation using an adaptive grid. The grid was obtained by minimizing a sum of squares of the independent and the dependent variables. The nodes are distrib-

uted creating an equally spaced mesh that is as nearly orthogonal as possible consistent with the solution surface.

Brackbill and Saltzman [3,4] developed a grid generation scheme designed to optimize a combination of smoothness, orthogonality, and cell volume. The measure of smoothness used in their scheme is given by

$$(15) \quad I_s = \int_D [(\nabla \xi)^2 + (\nabla \eta)^2] dv$$

and the orthogonality and cell volume measures, respectively, are given by

$$(16) \quad I_o = \int_D (\nabla \xi \cdot \nabla \eta)^2 J^3 dv$$

and

$$(17) \quad I_v = \int_D w J dv$$

In these expressions, (ξ, η) are the usual computational coordinates, J is the Jacobian of the transformation, and w is a weighting function used in the cell volume integral. A linear combination of these integrals is minimized in the form

$$(18) \quad I = I_s + \lambda_v I_v + \lambda_o I_o$$

where $\lambda_v > 0$, $\lambda_o > 0$ are undetermined multipliers.

In using minimization of Eq. (18) to develop an appropriate grid, the Euler equations must be derived. Even for a two space-dimension case these equations are formidable. While the Euler equations for typical two-dimensional problems are not repeated here, the details can be seen in the work by Brackbill and Saltzman. The Euler equations are solved in conjunction with the governing partial differential equations of the system to yield a solution for the entire problem.

Minimization of I provides a nice, elliptic grid generator assuring that certain smoothness, orthogonality, and error reducing properties are built in. There are some practical problems associated with this approach. For a truly adaptive grid, the weighting function in Eq. (17) must be solution dependent. At this time it is not clear what this function should be. The complexity and computing time requirements when a truly adaptive problem is attempted are also matters of concern. Simpler methods of obtaining an acceptable grid using minimization techniques are desirable. Clearly a less complicated measure of mesh quality is needed.

4. TYPE B METHODS. Type B schemes directly establish the grid speed term in Eq. (3). This grid speed is then integrated and the metrics of the transformation are calculated.

The most difficult part of constructing these methods is developing the rationale for the grid speed equation. Hindman et al. [6] derived a grid speed equation by using the time derivative of the Thompson scheme [13]. The Thompson scheme may be written

$$(19) \quad \begin{aligned} \nabla^2 \xi &= P \\ \nabla^2 \eta &= Q \end{aligned}$$

where the boundary point coordinates are given and the interior distribution is determined by the simultaneous solution of the system given in Eq. (19). The forcing functions (P,Q) are used to concentrate grid lines where desired. The system of equations governing the mapping is usually written using the physical coordinates as the dependent variables and may be written

$$(20) \quad \begin{aligned} G(x) &= 0 \\ G(y) &= 0 \end{aligned}$$

where

$$(21) \quad G = \frac{\alpha \partial^2}{\partial \xi^2} - \frac{2\beta \partial^2}{\partial \xi \partial \eta} + \frac{\gamma \partial^2}{\partial \eta^2} + \frac{1}{J^2} \left(\frac{P \partial}{\partial \xi} + \frac{Q \partial}{\partial \eta} \right)$$

with

$$(22) \quad \begin{aligned} \alpha &= x_\eta^2 + y_\eta^2 \\ \beta &= x_\xi x_\eta + y_\xi y_\eta \\ \gamma &= x_\xi^2 + y_\xi^2 \end{aligned}$$

and J again represents the Jacobian. If the time derivative of the transformation differential equation [Eq. (20)] is formed, a system of equations results and may be written

$$(23) \quad [S] \dot{\vec{z}} = \dot{\vec{r}}$$

where

$$(24) \quad \begin{aligned} \vec{z} &= (x_\tau, y_\tau)^T \\ \dot{\vec{r}} &= -\frac{1}{J^2} (P_\tau x_\xi + Q_\tau x_\eta, P_\tau y_\xi + Q_\tau y_\eta)^T \end{aligned}$$

and S is a matrix which includes spatial partial derivatives. The solution of Eq. (23) provides an expression for the grid speeds. In the initial work, P and Q were set equal to zero and numerous time-dependent solutions of fluid flow problems were computed. This technique was used with great success on a series of problems ranging from the classical inviscid supersonic blunt body problem to the diffraction of a shock wave passing over a ramp. Figures 4 and 5 present the physical domain and grid generated for these two cases. Unfortunately this technique is adaptive only in the sense of accommodating boundary point motion. If both P and Q are zero, no way of including the influence of the changes in the interior solution can be incorporated in the grid point motion. Recently, Hindman [7] has been computing flow field solutions using a grid generation scheme given by Eq. (23) where P and Q are functions of the computed solution. This creates an adaptive grid generator which includes the influence of both changes in boundary point location (moving boundaries) and changes in the interior solution.

Recently Rai and Anderson [10,11] and Anderson and Rai [2] have constructed an adaptive grid generator based upon a gravitational analogy. In order to demonstrate the basic idea, consider a one-dimensional problem with independent variables x and t . Since we obtain the solution by time integration of the partial differential equation, the grid speed is also easily integrated and the new grid positions obtained. In order to obtain the grid speed, we require the error, $|e|$, at each point and define an average error, $|e|_{ave}$, over all points. If $|e|$ is larger than $|e|_{ave}$ at a given location, we expect the local error in the solution to be reduced if more points are used. Likewise, if the difference between the local and average error is small, then fewer points are needed in a given region. Since the total number of points is fixed, this implies a contraction or stretching in certain areas of the physical domain. One also expects that the influence of one point on another diminishes as the distance between the two increases. With this in mind, the grid speed in computational space, ξ_t , may be written as

$$(25) \quad (-\xi_t)_i = K \sum_{j=i+1}^N \frac{|e|_j - |e|_{ave}}{r_{i,j}^n} - \sum_{j=1}^{i-1} \frac{|e|_j - |e|_{ave}}{r_{i,j}^n}$$

$$i = 2, 3, \dots, N-1$$

and the grid speed in physical space is given by

$$(26) \quad (x_t)_i = (-\xi_t)_i / (\xi_x)_i$$

In these equations, $r_{i,j}$ is the distance between points i and j in computational space, K is an empirical constant which must be adjusted to regulate the maximum grid speed, and n is a power which is adjusted to provide the desired radius of influence for a given point. The grid speed induced anywhere by a given point is proportional to the excess error at the point and is attenuated by the distance to that point raised to a power. The philosophy embodied in Eq. (25) may be interpreted as assuming that a numerical solution on a grid is the best when the error at each point in the grid is the same constant value.

An example demonstrating the application of this method is provided by the viscous Burgers' equation

$$(27) \quad u_t + uu_x = \mu u_{xx}$$

with initial conditions

$$(28) \quad u(0, x) = \begin{cases} 1 & x = 0 \\ 0 & 0 < x \leq 1 \end{cases}$$

and boundary conditions

$$(29) \quad u(t, 0) = 1$$

$$u(t, 1) = 0$$

The steady-state solution of this problem is given by

$$(30) \quad u = \hat{u} \tanh \left[\frac{\hat{u} R_e}{2} (1 - x) \right]$$

where

$$(31) \quad R_e = 1/\mu$$

and \hat{u} is a solution of the equation

$$(32) \quad \frac{\hat{u} - 1}{\hat{u} + 1} = \exp(-\hat{u} R_e)$$

Since the exact solution is known, the accuracy of the numerical calculation is precisely known, and the value of the adaptive grid scheme can be judged.

Figure 6 shows the solution error comparison for a fixed and an adaptive 11 point grid. In this example the error used in driving the grid was assumed to be u_ξ . While this does not correspond to the truncation error produced by using the second-order MacCormack method used in this calculation, a significant reduction in error is obtained using the adaptive grid. It is interesting to note that errors are reduced at the right side of the physical domain where large gradients exist while slight increases are observed at the left. This should be expected since the total number of grid points is fixed and the mesh spacing must increase at some points and decrease at others.

When the derivatives of the dependent variable are used to evaluate the $|e|$ terms required in the grid speed equation, they are formed using finite differences. To avoid noisy estimates, particularly when second or third derivatives are used, the solution must be smoothed before the derivatives are formed and used in the grid generator. Usually a three point average is sufficient although any smoothing operator will work.

This technique for creating an adaptive grid is easily extended to two dimensions. In this case the grid speed, ξ_t , is given by Eq. (25) with $|e|$ set equal to $|u_\xi|$ or higher derivatives of u , and the grid speed, η_t , given by a similar equation with u_η or higher derivatives used for the error estimate. The assumption that the grid speeds depend only upon errors in their respective directions provides for easy application of the method. As in the previous example, the contribution of each mesh point must be included, and as a result, the grid speed in either direction is obtained by summing over both i and j .

In the interest of brevity, details of the two-dimensional Burgers' equation calculation are not presented here. However a grid produced for the two-dimensional version of the example presented above is shown in Figure 7. In this case, the large gradient regions are at the right and upper part of the physical domain, and the adaptive grid shows a clustering of points in these regions. While error curves are not included, reductions similar to those calculated in the one-dimensional example are obtained. In addition to the simple examples presented here, a number of fluid dynamic problems have been solved. These include incompressible laminar boundary-layer flow, the supersonic inviscid blunt body problem, and the solution for supersonic inviscid flow over a pointed wedge with a detached shock wave.

In another recent paper, Rai and Anderson [12] have developed a technique for locally aligning the physical mesh with high gradient regions. This particular technique is most useful in computing solutions to hyperbolic systems of partial differential equations which include surfaces of discontinuities in the dependent variables. The applications of the original technique were to problems involving shock waves in high speed fluid flow and that development will be presented here.

The presence of shock waves in supersonic flow creates problems for the computational fluid dynamicist because shocks represent discontinuities in the dependent variables when the inviscid equations of motion are considered. Lax [9] showed that shock waves could be "captured" as part of the solution using no special treatment if the conservative form of the governing partial differential equations is used. The usual conservation-law form of the inviscid equations for a steady supersonic flow is

$$(33) \quad \frac{\partial \vec{E}}{\partial x} + \frac{\partial \vec{F}}{\partial y} = 0$$

and

$$\vec{F} = \vec{F}(\vec{E})$$

where both E and F are vectors. When shock waves are captured using finite-difference methods, the solution usually oscillates at shock waves because of the discontinuous nature of the dependent variables. Since a solution with a shock wave mathematically represents a weak solution of Eq. (33), the condition which must be satisfied at the shock may be written (see Witham [14])

$$(34) \quad [\vec{E}] \cos \alpha_1 + [\vec{F}] \cos \alpha_2 = 0$$

where the square brackets represent the jump in the function across the discon-

tinuity, and $\cos\alpha_1$ and $\cos\alpha_2$ are the direction cosines of the normal to the discontinuity with respect to the x and y axes. Figure 8 identifies the normal and the angles α_1 and α_2 . If an adaptive grid is used, which aligns with the shock in such a way that $\alpha_2 = 90^\circ$, the remaining condition that must be satisfied becomes

$$(35) \quad [\vec{E}]\cos\alpha_1 = 0$$

Since $\cos\alpha_1 \neq 0$, the jump in \vec{E} is zero which simply requires that \vec{E} be continuous across the shock wave. A scheme which aligns one of the coordinates with the shock wave can be constructed. If a series of points in a grid is considered as in Figure 9, a scheme which causes mesh alignment with high gradient regions results if the grid speed at a point C is given by

$$(36) \quad \xi_{tC} = \frac{K|h_\xi|_0|h_\eta|_0(-1)^k}{r_{OC}^n}$$

where

$$k = \begin{cases} 1, & \text{sgn}(h_\xi/h_\eta)\text{sgn}(\eta_0 - \eta_C) < 0 \\ 2, & \text{sgn}(h_\xi/h_\eta)\text{sgn}(\eta_0 - \eta_C) > 0 \end{cases}$$

and h is any flow variable such as pressure or density which is used to identify high gradient regions. The grid point speed at any point C is determined by the gradient of this flow variable. The effect of this is to produce a rotation of the line segments until they become locally parallel to the maximum gradient lines.

To demonstrate the effectiveness of this technique, a smooth function with a very high gradient region was used to drive the grid. Since the gradient information, h_ξ , h_η , is known analytically, and the high gradient area is also known, this provides a good test of the shock aligning scheme. Figure 9 demonstrates the shock alignment scheme for such a unit problem. The dark area is the zone where high gradients exist while the h function is constant in the rest of the domain. The alignment of the grid in the high gradient region is apparent. It is also important to notice that grid alignment is a very local effect.

Another demonstration of the effectiveness of a shock aligning grid is given by the calculation of a flow field due to a straight oblique shock in a uniform supersonic freestream. Figure 10 shows the shock wave location and shows the position of the fixed grid also used for a comparison. The flow is from the top of the figure toward the bottom at a freestream Mach number of 2.0 with a shock wave angle of 50° . The solution to this problem was obtained numerically as the time asymptotic limit of an unsteady flow. The two-dimensional time-dependent equations of motion were solved in conjunction with the grid speed given in Eq. (36), and the resulting mesh and shock location are shown in Figure 11. The nearly perfect alignment of the shock and the grid are apparent. Figures 12 and 13 show the pressure through the mesh at $y = 0.208$ and 0.0 ,

respectively. Again the effect of the alignment is clear. No oscillations occur in the grid aligned results while the usual dispersive behavior is evident when a standard fixed grid is used.

5. CONCLUDING REMARKS. A number of adaptive grid techniques which are currently being used or have the potential for use in solving practical problems have been reviewed. These methods include schemes which directly determine either physical point location or transformation metrics (type A) and those techniques which directly provide the grid speeds (type B). Techniques used to develop these methods range from variational approaches to ad hoc schemes which largely employ physical intuition.

Having reviewed some of the more promising methods for generating adaptive grids, a few comments on the direction of future work seem appropriate. The direction of future work as perceived by any researcher in this area depends upon that person's experience and the constraints placed upon his efforts. The engineer or scientist attempting to solve practical problems, using numerical methods, does not have the luxury of either unlimited computer time or the use of an infinite number of mesh points. We should rule out those schemes which require excessive time to use and support techniques which are more economically employed.

Among those schemes reviewed in this paper, the methods where a functional minimization is employed become unduly complex when the Euler equations are solved in conjunction with the original partial differential equation. The time required to generate the mesh may be larger than or at least a large fraction of the time required to solve the original equation. These methods do have a firm mathematical basis and permit one to exercise positive control over those elements included in the definition of grid quality employed. For this reason, work using this approach is valuable.

The techniques which provide the grid speed directly have a definite advantage because they are easily used in multidimensional problems. Since the goal of most investigators is the construction of methods for use in three-dimensional problems, this seems to be a definite plus. The disadvantage is that these techniques are approximate methods largely based upon intuition. Even though the simple grid speed methods reviewed in the last section are error reducing, a better foundation justifying their use is needed.

Perhaps the next generation of adaptive grid schemes will result from studies using minimization techniques. These studies can be used to construct simplified approaches which yield nearly the same results but are much simpler to implement and more economical to use.

One of the areas deserving a concerted effort is that of defining grid quality. Grid quality treated in adaptive grid work is usually concerned with improved resolution of some physical event or with reducing errors in the solution. The area of error estimation when using finite-difference techniques is very important. Clearly, if better error estimates are available, better grid systems can be generated.

As a final comment, the adaptive grid field is new. Everyone should be encouraged to explore new ideas for generating adaptive grids with the goal of introducing better techniques for computing solutions to partial differential equations.

The support of this work by NASA under Cooperative Agreement NCCI-17 is gratefully acknowledged.

1. C.M. ABLOW & S. SCHECHTER, "Generation of boundary and boundary-layer fitting grids," *Numerical Grid Generation Techniques*, NASA Conference Publication 2166, October 1980, pp. 121-128.
2. D.A. ANDERSON & M.M. RAI, "A new approach to solution adaptive grids," in *Computers in Flow Predictions and Fluid Dynamics Experiments* (Papers at the winter meeting of the ASME, Washington, 1981), ASME, New York, 1981.
3. J.U. BRACKBILL & J. SALTZMAN, "An adaptive computation mesh for the solution of singular perturbation problems," *Numerical Grid Generation Techniques*, NASA Conference Publication 2166, October 1980, pp. 193-197.
4. J.U. BRACKBILL & J. SALTZMAN, "Adaptive zoning for singular problems in two dimensions," Los Alamos Scientific Laboratory, LA-UR-81-405, 1981.
5. H.A. DWYER, R.J. KEE & B.R. SANDERS, "An adaptive grid method for problems in fluid mechanics and heat transfer," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, July 1979, pp. 195-204.
6. R.G. HINDMAN, P. KUTLER & D. ANDERSON, "Two-dimensional unsteady Euler-equation solver for arbitrary shaped flow regions," *AIAA Journal*, v. 11, April 1981, pp. 424-431.
7. R.G. HINDMAN, in private communication, 1982.
8. G.H. KLOPPER & D.S. McRAE, "The nonlinear modified equation approach to analyzing finite-difference schemes," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, June 1981, pp. 317-333.
9. P.D. LAX, "Weak solutions of nonlinear hyperbolic equations and their numerical computation," *Comm. Pure and Applied Math*, v. 7, 1954, pp. 159-193.
10. M.M. RAI & D.A. ANDERSON, "Grid evolution in time asymptotic problems," *J. Computational Phys.*, v. 43, 1981, pp. 327-344.
11. M.M. RAI & D.A. ANDERSON, "Application of adaptive grids to fluid flow problems with asymptotic solutions," Presented at the AIAA 19th Aerospace Sciences Meeting, AIAA Paper 81-0114, January 1981.
12. M.M. RAI & D.A. ANDERSON, "The use of adaptive grids in conjunction with shock capturing methods," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, June 1981, pp. 156-165.
13. J.F. THOMPSON, F.C. THAMES & C.M. MASTIN, "Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies," *J. Computational Phys.*, v. 15, 1974, pp. 229-319.
14. G.B. WHITHAM, *Linear and Nonlinear Waves*, Wiley, New York, 1974.
15. N.N. YANENKO, E.A. KROSHKO, V.V. LISEIKIN, V.M. FOMIN, V.P. SHAPEEV & Y.A. SHITOV, "Methods for the construction of moving grids for problems of fluid dynamics with big deformations," *Lecture Notes in Physics*, v. 59, 1976, pp. 454-459.

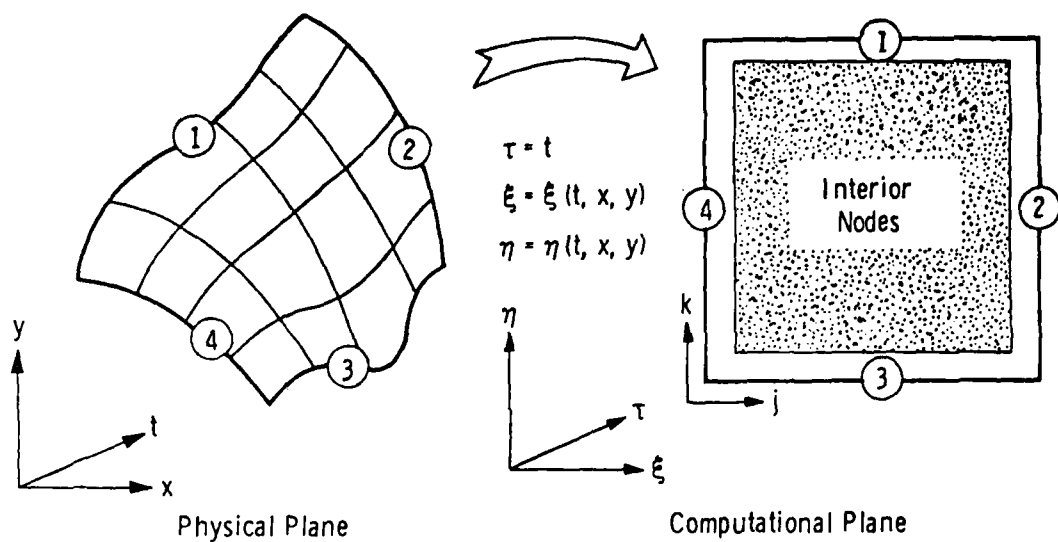


Figure 1 Transformation to computational space

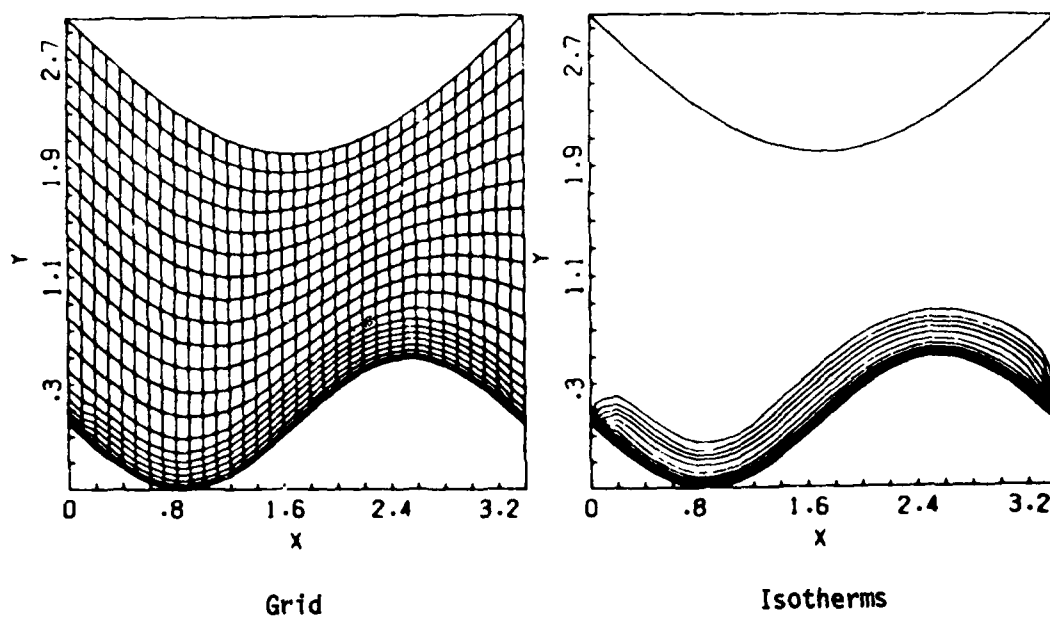
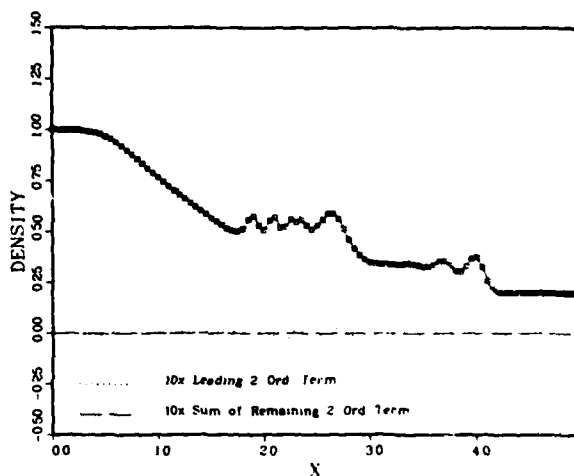
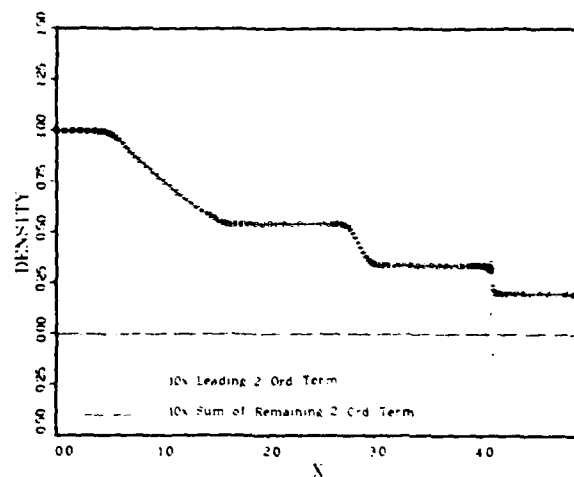


Figure 2 Results for one-dimensional heat conduction problem of Dwyer et al. [5]



Uniform Grid



Adaptive Grid

Figure 3 Shock tube results of Klopfer and McRae [8]

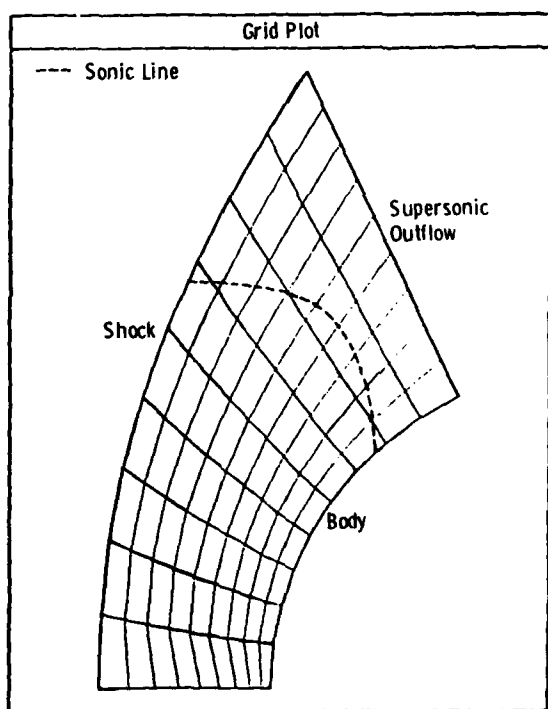


Figure 4 Grid computed for the blunt body problem

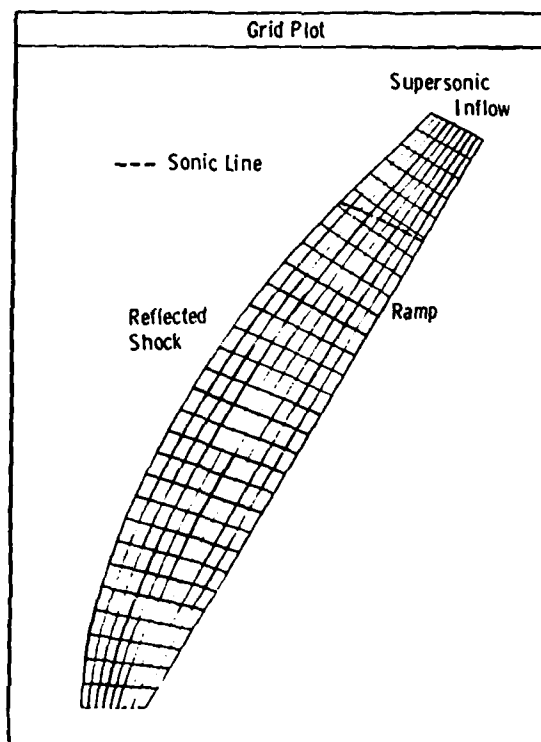


Figure 5 Grid generated for blast diffraction problem

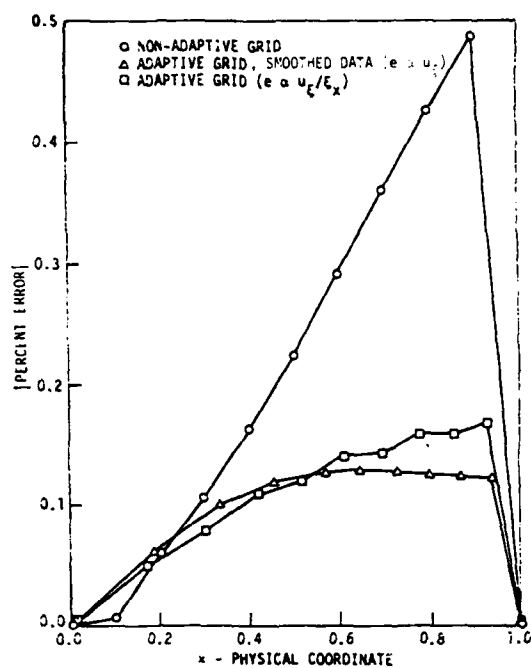


Figure 6 Error curves for Burgers' equation

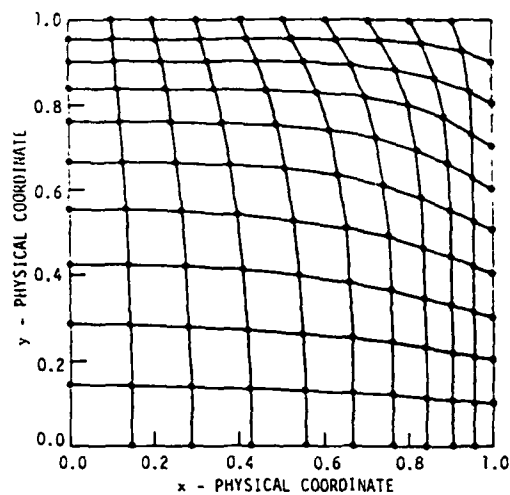


Figure 7 Adaptive grid for two-dimensional Burgers' equation

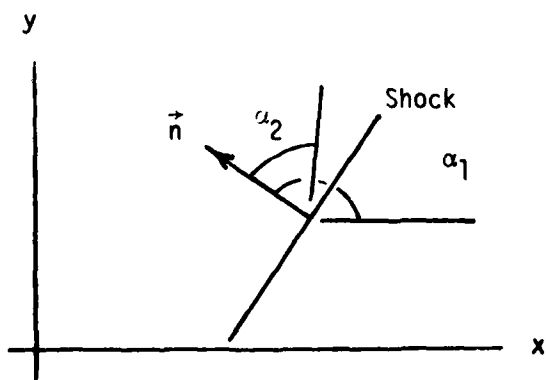


Figure 8 Geometry of shock normal orientation

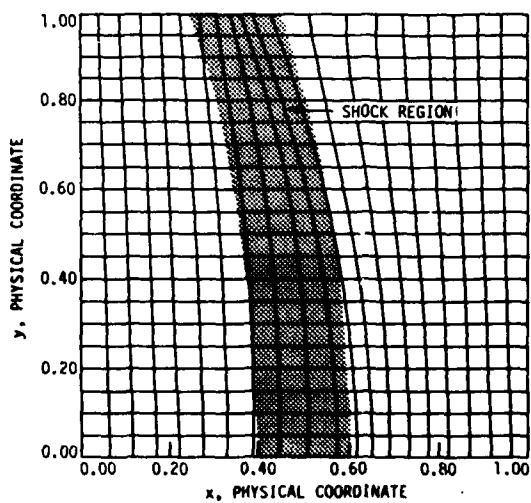


Figure 9 Demonstration of shock aligned grid

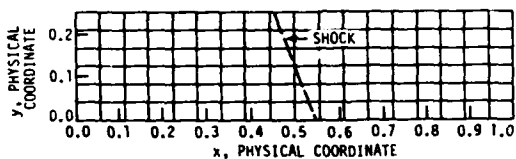


Figure 10 Fixed grid for supersonic flow with shock wave

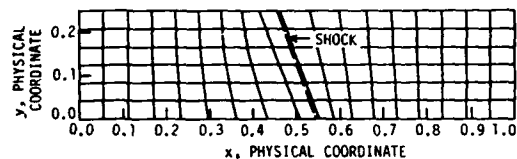


Figure 11 Shock aligned grid for supersonic flow problem

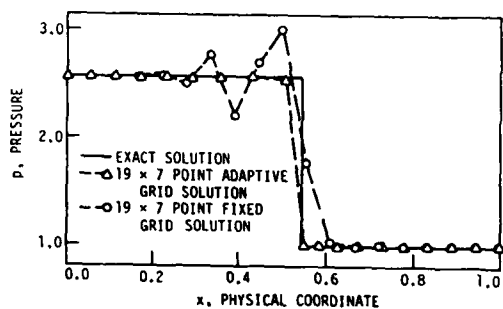


Figure 12 Pressure comparison showing effectiveness of shock aligned grid at $y = 0.208$

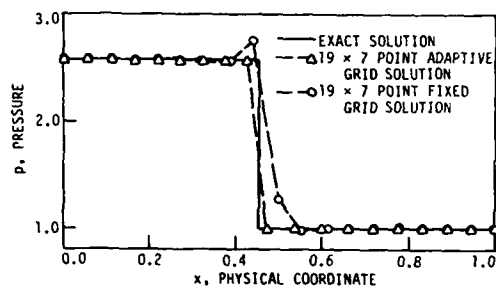


Figure 13 Pressure comparison at $y = 0.0$

INTERACTIVE DESIGN OF LASER ELECTRODES USING
ELLIPTIC GRID GENERATION AND SEMIDIRECT/MARCHING METHODS

Patrick J. Roache
Ecodynamics Research Associates, Inc.
P. O. Box 8172
Albuquerque, New Mexico 87198

ABSTRACT. This paper describes a computational effort to develop computer codes for rapidly and accurately modeling the electric fields within laser cavities. Semidirect/marching methods are used both for the generation of two-dimensional boundary-fitted grids using the elliptic generating equation approach, and for the solution of electric field problems in those coordinate systems. The efficiency of the semidirect/marching methods makes possible interactive design of the laser electrodes using a modest computer. Also described are techniques for high-order accuracy, a method for precise grid control at interior points, and applications to the elliptic grid generation problem of computer Symbolic Manipulation.

1. INTRODUCTION: THE LINEAR AND NONLINEAR LASER PROBLEMS. The objective of the computational effort described herein was to develop computer codes for rapidly and accurately modeling the electric fields within laser cavities. These codes should be fast enough to make the interactive design process practical, and accurate enough to resolve the maximum electric field, which is an important limiter of the power output. The designer should be able to perturb the laser operating parameters and/or the electrode geometry, and quickly obtain new solutions.

Both the linear and nonlinear electric field problems are of practical interest to various laser concepts. Our original efforts were directed towards the nonlinear problem in pulsed electric lasers. In the design of electron beam lasers, it is desirable to have a nearly uniform energy deposition throughout the cavity. This energy deposition is governed by the solution of the nonlinear elliptic equation for electric potential ϕ , given by

$$\nabla \cdot \sigma \nabla \phi = 0$$

where the conductivity σ is a nonlinear function of the electric field $E = \nabla \phi$. (For the linear problems, σ is constant.) The solution of this equation for a reasonable grid resolution in two dimensions is a time consuming effort using conventional methods (ref. 1). For our initial studies, the ionization S of the external electron beam gun was modeled empirically, following ref. 2, by the following equation.

$$S = \exp(-\gamma x) \operatorname{atan} \left(\frac{y+a}{x} \right) - \operatorname{atan} \left(\frac{y-a}{x} \right)$$

where $\gamma = E/2 \cdot V$. The electron beam has a voltage V and a width $2a$ located at

$x = 0$ between $y = -a$ and $y = +a$. From the same reference, the nonlinear conductivity σ is given by

$$\sigma = C \cdot E^{0.45} S^{0.5}, c = 0(1)$$

The laser cavity was first modeled in a cartesian coordinate system with straight electrodes. In this system, we could obtain completely converged nonlinear solutions in approximately 5 second on a 31 x 31 grid using a time-shared CDC 6600 Computer. This is two orders of magnitude faster than the computer time necessary to solve the problem using a triangular finite element system.

We have also used other nonlinear relations for conductivity σ , and are currently working on the use of Monte Carlo calculations for σ . Although the nonlinearity adds difficulty, the more significant problem arises when the designer attempts to solve the problem with general electrode shapes.

We immediately dismissed the approach of fitting a general boundary shape into a cartesian grid using "partial cell" formulas, for reasons of accuracy. Since the quantity of interest is a derivative of the solution, and since its maximum value occurs on the surface (always, for the linear problems) it was clear that "partial cell" formulas would not provide sufficient accuracy. A boundary-fitted nonorthogonal coordinate system was the obvious choice.

2. ELLIPTIC GRID GENERATION BY SEMIDIRECT METHODS. We adapted the approach pioneered by Thompson et al. (ref. 3), using elliptic generating equations to construct the nonorthogonal grid. Two equations for the new coordinates ξ and η are first written in the "physical" or original coordinate system (usually cartesian) where we know the form of the equations governing the electric field. The two equations for $\xi(x,y)$ and $\eta(x,y)$ are linear but have the same difficulty as the original problem, i.e. solution in x and y would require "partial cell" formulas. This is avoided in the Thompson approach by reversing the dependent and independent variables. All calculations are now done in the simple cartesian coordinate system in the transformed plane (ξ,η) , but the transformed equations are now nonlinear (quasilinear).

The two coupled nonlinear equations are solved in the transformed plane (ξ,η) for the physical coordinates (x,y) .

$$L(x) = 0, L(y) = 0$$

where, for $e = x$ or y ,

$$L(e) \equiv \alpha e_{\xi\xi} - 2\beta e_{\xi\eta} + \gamma e_{\eta\eta}$$

The coefficients are nonlinear functions of x and y . See Thompson, et al. (ref. 3) for details, and for the use of additional nonhomogeneous terms P and Q for coordinate adjustments.

In customary usage, these equations are solved by point or line iterative methods, which are usually slow. In our work, we use semidirect/marching methods to solve the grid equations, and then to solve the electric field equation in that grid.

3. SEMIDIRECT/MARCHING METHODS: THE GEM CODES. Semidirect methods are rapid finite difference methods for solving various steady-state and slowly varying time-dependent nonlinear problems. Fast elliptic solvers are used to solve linearized equations directly, which are then iterated to solve the nonlinearity. Applications of semidirect methods to problems, many in fluid dynamics, are given in ref. 4. For the nonseparable partial differential equations of interest here, the fast elliptic solver used is some variation of marching methods for elliptic equations. The algorithms involved have been described in detail (ref. 5) and timing and accuracy tests of a particular software realization of the marching methods, called the GEM codes, have been reported (ref. 6).

Although stabilizing schemes exist (ref. 5,6), as a practical matter, the marching methods depend on a favorable cell aspect ratio $\Delta\xi/\Delta\eta$ to stabilize the inherently unstable spatial marching procedure. They are thus well suited to problems with a grid refinement in one coordinate in the transformed plane. *The marching methods are not suitable for problems in which there is a significant grid refinement in both coordinate directions in the transformed plane.* However, for many practical problems, they are very well suited.

The advantages of the marching methods are their generality and speed. Unlike "fast Poisson solver" algorithms such as FFT or odd-even reduction, the marching methods (1) do not depend on separability of the coefficients, and (2) can treat the 9-point operator directly, even for nonseparable stencils. Both these advantages are pertinent to nonorthogonal grid problems, not only in the solution of the grid by elliptic pde's, but also in the solution of the physics (in the present case, the electric field equations) no matter how the non-orthogonal grid is generated. As for speed, the marching methods will initialize in order (M^3) operations for an $M \times M$ cell problem, and will solve repeat solutions in the optimal order (M^2) operations. For large two-dimensional problems using a 5-point operator, repeat solutions by actual timing tests are obtained (ref. 6) in the equivalent of 2 point SOR iterations including convergence testing.

4. APPLICATION TO ELLIPTIC GRID GENERATION EQUATIONS. The semidirect/marching methods are particularly well suited to the solution of the elliptic grid generating equations, provided that the cell aspect ratios are favorable. In our semidirect approach, the two equations are first linearized about some initial guess for the grid, giving values of α^0, β^0 , etc. We then solve a sequence of linear problems, indicated by

$$L^0(e^k) = S(e^{k-1})$$

where L^0 is based on the initial guess α^0, β^0 , etc. and S is defined by

$$S(e) \equiv - \{ (\alpha - \alpha^0) e_{\xi\xi} - 2(\beta - \beta^0) e_{\xi\eta} + (\gamma - \gamma^0) e_{\eta\eta} \}$$

If immediate updating of the coefficients were used (true Picard method), the coefficients in L^0 would be re-evaluated at each iteration and the GEM solution would be reinitialized, requiring order (M^3) operations for each iteration. Instead, we attempt a single initialization of the GEM code using a quasi-Picard method. Depending on the adequacy of the initial guess, this single initialization may be adequate, or we may require reinitialization during the solution process. The decision to reinitialize is automated and is based on the requirement for at least an 80% reduction in the maximum change in x and y at each iteration. Also, in the iterative design process, the initialization from a previous design (i.e. a previous laser electrode geometry) can be used for the next grid generation.

The semidirect/marching methods are well suited to this problem of elliptic grid generation for two reasons. First, although two coupled nonlinear equations are used, there is only one matrix for the two equations. Thus, only one matrix initialization is used, and only one set of coefficients must be stored. Second, although the equations are nonlinear and coupled, they are not coupled in the boundary conditions. This adds to the speed of the iterative convergence process. (For the Navier-Stokes equations, the coupling of the boundary conditions leads to time-like iterative behavior, which is comparatively slow; e.g. see refs. 4,7.)

5. ACCURACY AND TIMING TESTS. For moderate geometries, the semidirect/marching methods give solutions for the grid in typically 8 to 10 iterations, requiring less than 4 seconds on a CDC 6600 for a 31×31 grid with poor initial guesses. We use an unusually tight convergence criterion of $\delta x, \delta y < 10^{-5}$, because we are interested in using Richardson extrapolation to fourth order accuracy for the solutions of the physics equations; this requires no oscillations in the solution for either the coordinate system or the physics equations (ref. 8). The number of iterations required is not a strong function of grid size, and the marching error is tolerable for most problems encountered so far (of the order 5×10^{-6} for a 31×61 grid). As yet, we have had no experience with coordinate system control using the P and Q terms (ref. 3). Fortunately, many geometries of practical interest to the electrode design area are convex in the region of most interest and do not require additional coordinate control. The present code is being used for interactive computer design of several laser systems.

The electric field solutions are also obtained with the semidirect/marching methods once the coordinate system has been generated. For linear field equations with 1-point or 2-point derivative boundary conditions, the equations are solved directly. For the nonlinear field equations and for 3-point derivative boundary conditions, iteration is required. A representative problem is solved in the order of 10 iterations, requiring less than 5 seconds on a CDC 6600. However, we have encountered nonlinearities in σ which required 50 iterations.

The linear problem is of practical interest, and has been used as an accuracy test by comparison of the computed results with those of the Rogowski electrodes, obtained by conformal transformation methods. With boundary points equidistributed in arc length, we predict the E-field to plotting accuracy in a 25×25 grid. Using a distribution of boundary points weighted by surface curvature, we have obtained plotting accuracy in a 13×13 grid.

It appears that a good multigrid code using nonlinear grid interpolation (FAS) can achieve the same level of efficiency as the semidirect/marching methods for the nonlinear problems (ref. 9). For the linear problem, the marching methods as embodied in the GEM codes are the fastest. However, they are limited in resolution to about a 100x100 grid with favorable cell aspect ratios. More importantly, they are attractive in 3 dimensions only for problems which are separable in the third coordinate so that a FFT can be used (ref. 5). The marching methods appear to vectorize well, especially for repeat solutions, for the 5-point operator. On a vector machine, 9-point operators would be best treated iteratively by lagging, as is customarily done with linear iterative methods. The vectorizing of multigrid codes is an open question at this time. The comparison of marching methods, multigrid methods and the simpler fully vectorizable iterative methods (such as hopscotch SOR) on vector machines will be a complicated job, dependent on the particular machine architecture, the problem size, and the coding details. We intend to include options for the use of various solvers in our laser codes in the near future.

6. CONTINUATION METHODS FOR DIFFICULT GEOMETRIES. Good initial conditions for the grid can be a problem, whether the grid generating equations are solved by semidirect/marching methods or by more conventional iterative methods. Particularly, for slit-like geometries, initial conditions obtained by simple interpolation in the transformed plane can give crossed coordinate lines and negative Jacobians, which can prevent iterative convergence of the nonlinear problem.

We have developed two continuation methods for this problem. Both attain the final solution in N continuation steps (where N is selected by the code user). The weighting function W varies from 0 to 1 for the sequence of problems,

$$W = 0, 1/N, 2/N, \dots, (N-1)/N, 1.$$

The first continuation method builds up to the true boundary conditions. With $B = x$ and y boundary conditions, the continuation method is

$$B^k = (1-W)B^0 + W \cdot B^{\text{true}}$$

where B^0 is some trivial initial geometry, such as a rectangle.

The second method builds up to the true generating equations, and was suggested by Maliska's work (ref. 10) using point SOR for the solution. The coefficients α , β , and γ are built up from

$$\beta^k = W \cdot \beta^{\text{true}}, A^k = (1-W) + W \cdot A^{\text{true}}$$

where

$$A = \alpha \text{ and } \gamma.$$

This starts from the linear, decoupled problem

$$x_{\xi\xi} + x_{\eta\eta} = 0$$

$$y_{\xi\xi} + y_{\eta\eta} = 0$$

We have had success with both methods, but the second is preferable. It is more systematic, and avoids some clumsy scaling problems of the first. For a rather severe slit-like laser geometry, only two continuation steps were required to solve the grid. (Note that for very mild problems, this first continuation step might produce an adequate grid, and could utilize any fast Poisson solver for separable equations.)

7. SOLUTION OSCILLATIONS NEAR GRADIENT BOUNDARIES. An illuminating behavior arose in the application of symmetry boundary conditions to the electric field equations. For symmetry at $\xi = 0$, the transformed equation requires

$$\frac{\partial \phi}{\partial \eta} = (\alpha \phi_{\xi} - \beta \phi_{\eta}) / J \alpha^{\frac{1}{2}} = 0, \text{ where } J = \text{Jacobian.}$$

The marching code GEM requires one-sided differences for ϕ_{η} because the boundary conditions must be separable in the march direction. Depending on the curvature at the boundary (the sign of β) and the march direction, this can be analogous to *downwind* differencing along the boundary, and can produce oscillations in the solution of the physics equations. In analogy with the well-known fluid dynamics problems, we would anticipate that other workers may have encountered this behavior using centered differences for ϕ_{η} .

The cure, which almost certainly has been applied in practice elsewhere although not reported (nor perhaps recognized) is to have a nearly orthogonal grid near symmetry and other gradient boundaries, giving $\beta \approx 0$. (One could also set $\beta = 0$ by reflection (ref. 10) but this gives a discontinuity in the grid which will slow the truncation error convergence.)

In the GEM solutions, true second-order accuracy is obtained by a deferred correction approach, lagging the difference between the one-sided and centered forms for ϕ_{η} . It is even more robust, for geometries in which β might change sign along the boundary, to lag the entire ϕ_{η} , along with the deferred correction for the 3-point ϕ_{ξ} and any nonlinearities, and this is now our standard procedure. Note, however, that the GEM code now cannot be considered a direct solver for gradient boundary conditions in a nonorthogonal grid; this is a code limitation, since marching methods (i.e. the algorithm) can be adapted to solve this problem directly.

8. TECHNIQUES FOR HIGH ACCURACY SOLUTIONS. For the laser design problems which we have encountered to date, moderate accuracy has been quite adequate. (The more important problem has been resolution of the peak E-field, and we expect to soon be working on a solution-adaptive grid generation method to address this problem.) For other applications, high accuracy may be desirable. Obtaining high accuracy solutions in boundary-fitted coordinates requires special comment.

Obtaining high accuracy solutions generally involves using high-order discretization and/or systematic grid refinement. There are remarkably few published studies of strongly multidimensional problems which do a convincing job of establishing accuracy, even for problems defined in cartesian coordinates. For general nonorthogonal coordinate transformations, we need a systematic method for refining this mesh and assuring smoothness of the mesh. This

requirement is satisfied by Thompson's elliptic generating approach (and by simple analytic stretches, etc.).

We consider here three techniques for obtaining high-order solutions in general nonorthogonal grids; Richardson extrapolation, the direct use of high order equations, and deferred corrections.

Richardson extrapolation must be applied with great care. Incomplete iteration noise and machine round-off error will be magnified by the extrapolation, and the enhanced order of accuracy will not occur near boundaries unless consistently-ordered discretizations are used at boundaries. However, when carefully implemented (ref. 8) this technique does give high order accuracy in nonorthogonal coordinates, and is not troubled at all by the cross-derivatives. It gives $O(\Delta^4)$ accurate solutions only on the subgrid of the finest grid calculated, however. It may be possible to obtain the $O(\Delta^4)$ solution on the finest grid by interpolation; this approach remains to be worked out and verified.

The direct use of high-order equations, either conventional or "compact" stencils, gives high order solutions on the full grid. However, there is trouble in formulating stencils for cross-derivatives and near-boundary points in nonorthogonal coordinates. Also, the iterative solution methods may deteriorate with high-order stencils. (They should never be used directly in marching methods, since the effect on the march stability is disastrous; see ref. 5.) The deferred correction technique avoids this latter difficulty, and further provides a convenient measure of truncation error convergence. However, the difficulties with the cross-derivatives remain.

To our knowledge, ref. 8 (utilizing Richardson extrapolation) presents the only multidimensional $O(\Delta^4)$ solution in a nonorthogonal boundary-fitted grid.

9. PRECISE COORDINATE CONTROL AT INTERIOR POINTS. It is often desirable to precisely control the position of grid nodes at some interior points. In laser calculations, the electric field can be affected by the presence of dielectric materials in the cavity, and calculation accuracy could be enhanced if the grid points were on dielectric boundary. Such precise control is simply achieved by algebraic (e.g. ref. 11) and ad hoc grid generation methods. In the elliptic generating technique, the "tuning" of the nonhomogeneous terms P and Q as in ref. 3 provides considerable adjustment of the grid, but not precise control.

Precise placement of grid points is easily achieved by partitioning the grid solutions along the desired interior boundary. This can be implemented either by patching separate solutions together, or by locally defining the discretized problem to be the identity equation, i.e. replacing the 9-point stencils for the x and y differential equations by

$$x = x_{ib} \text{ and } y = y_{ib},$$

where "ib" refers to the desired interior boundary. Patching results in a timing and storage penalty in the GEM codes, but the patch line can also be used to stabilize the march (refs. 5, 6). The second implementation will not

work with the GEM codes, as it results in a singular matrix for the marching procedure. Either implementation speeds convergence for point and line iterative methods.

However, precise placement of grid points is not the real difficulty; rather, it is achieving a smoothness of the grid through the interior boundary. Smoothness will be possible only if (1) the angles of the coordinate lines passing through the interior boundary, and (2) the grid spacings along those coordinate lines, are "equal" (to some discretized measure) on both sides of the boundary. This can be accomplished with Steger and Sorenson's algorithm (refs. 12, 13) which iteratively adjusts the nonhomogeneous terms P and Q so as to achieve the user-specified coordinate spacing and angle at boundaries. It will also be possible to slide the grid points along the interior boundary following some solution-adaptive scheme. The position of the interior boundary itself can likewise be adjusted to follow the physics solution, e.g. the dividing streamline in separated flow.

As an alternative to Sorenson's algorithm, we can achieve smoothness by specifying gradient rather than Dirichlet boundary conditions in the solution of $x(\xi, \eta)$ and $y(\xi, \eta)$. This is an entirely different algorithm and will generate a different grid. Unfortunately, this formulation nonlinearly couples both the x and y boundary derivatives in both the ξ and η directions. This is expected to slow convergence in the semidirect formulation (ref. 4) and in point iterative solutions, especially if solution-adaptive procedures are simultaneously used. Actually, a weighted combination of Dirichlet and both derivative conditions (generalized Robbins' condition) is the obvious candidate. A comparison of these two approaches will be undertaken in the near future; both can be extended to 3D.

10. SENSITIVITY TO CROSS DERIVATIVES. We have generally been impressed with the difficulty of code verification for general nonorthogonal coordinate problems. In particular, the experience related here violated out intuition on the sensitivity of the solutions to the cross derivative terms like $x_{\xi\eta}$, $\phi_{\xi\eta}$ etc. The experience arose from a coding error in which the cross derivative terms were all calculated a factor of 2 larger than correct. The error was not detected early because the solutions looked good for mild but non-trivial geometries. For electrodes in a quadrant where the lower electrode was described by a $\cos^{1/2}$ curve and the upper electrode by $\cos^{1/4}$, the grid generated and the solution for the E-field were quite accurate. Likewise, the solution for the Rogowski electrode differed by only 0.4% from the exact ϕ , using only a 13x13 grid. However, in systematic convergence testing (performed by H. Happ of Tetra Corporation), the error did not reduce as the grid was refined. The coding error was detected and corrected, and the previous cases were recalculated. The factor of 2 error in the cross derivatives proved to affect the coordinate generation by less than 0.01% in the location of any x and y of the grid nodes, and to affect the E-field (derivative of the ϕ solution) by 0.016%. The conclusion might seem obvious, that the solutions are very insensitive to the cross derivatives. However, this is actually quite problem dependent. For a slit-like geometry, the coding error seriously affected the grid generation. Iterative convergence was obtained only with the extreme of 20 continuation steps plus the use of extensive under-relaxation of boundary and interior points. The resulting "mesh" was a mess, with coordinate lines

that crossed and extended outside of the physical domain, violating the maximum principle. When the coding error was corrected, the method converged to a perfectly good grid in 2 continuation steps. For this class of problems, we conclude that the grid generation process is highly sensitive to the cross-derivatives. Aside from coding errors, this experience also seems to bear on the robustness of alternate elliptic generating systems which use simpler equations in the transformed plane; their chances of success for difficult geometries appears to be poor.

11. SYMBOLIC MANIPULATION AND GRID GENERATION. Coding errors such as the one described above plague all computational work, and the chance for error increases as the complexity of the problems increase. As noted above, we have been impressed with the difficulty of code verification for the transformed grid problems. We have also been impressed with the complexity of the 3-dimensional equations for general nonorthogonal grids.

In association with Prof. Stanly Steinberg of the University of New Mexico, we are addressing this and related problems using computer Symbolic Manipulation. These are not floating-point calculations, but symbolic operations, e.g. the chain rule differentiation, performed by computer logic. The gathering of coefficients is likewise done symbolically, as is the actual writing of the Fortran subroutines to define the problem. The symbolic code used is a VAX computer version of the code MACSYMA developed over many years at the MIT Lincoln Laboratories.

To recapitulate: we are using MACSYMA to (1) analytically generate the transformation equations, and (2) to actually write a Fortran subroutine to produce the 9-point stencil defining the matrix problem.

Once the computer has written the subroutine defining the problem, the coefficient matrices defining the stencil are passed to some canned solver, in this case the GEM codes. Both the grid generation problem and the physics equation are solved the same way. Except for input/output and processing of the results, as well as the passing of the matrix problem to the canned solver, the user obtains the answer without writing Fortran or similar code.

The general second-order two-dimensional equation has been solved in this manner, and the results verified by comparison to the hand-coded coefficient matrices. The analytic generation of the transformation equations and the writing of the Fortran subroutine require about 10 minutes on a VAX 780. The three-dimensional problem has also been solved, but the computer time increases dramatically due to the computational complexity of the chain rule operations, similar to the classic "sorting" problem. We are currently involved in the code verification. Rather than generate a hand-coded version, we will obtain three-dimensional solutions of the algebraic equations (using a hopscotch SOR "canned" solver) and verify the code by convergence testing to the exact solution of highly stretched coordinate problems.

In the near future, we intend to work on the relatively straight-forward problems of multiple equations, higher order equations, perturbation terms in the source term formulated so as to give deferred corrections to higher order accuracy and/or nonlinear terms, and validation of all these.

More difficult problems are conservation forms, upwinding (or other conditional differencing), complicated boundary conditions (currently we have used only Dirichlet conditions), and optimization. It is likely that the Fortran code generated will always be less efficient than what could be obtained with expert hand coding. This situation is viewed as analogous to the situation of efficiency attainable from high-level languages like Fortran vs. assembly language. The "efficiency" sought is not that measured by CPU seconds for code execution, but by calendar years for code development.

Human errors are still possible in this process, but they are a different level of error. Grand mistakes will occur, but not the petty ones of writing $S(I+1,J)$ when the term should have been $S(I-1,J)$, etc.

The following areas of application for Symbolic Manipulation appear most promising.

(1) Combination of perturbation methods and numerical methods. These "semianalytic" approaches have already been used with some success, and are not difficult for regular perturbation problems. With insight, they can be used for singular perturbation problems, and could be used in general grid problems to remove grid-introduced singularities.

(2) Coordinate transformations, especially in conjunction with (3).

(3) Constitutive equation testing, in areas like turbulence modeling, non-Newtonian fluids, soil mechanics, gravitational theory.

(4) Generation and analysis of new discrete forms via finite difference, finite element, least squares, etc. methodologies.

The prospect of virtually error-free testing of constitutive equations and difference forms is most attractive. I predict that the use of Symbolic Manipulation in these and other problems will shortly be recognized as the way of the future, and that the practice of disciplines like computational fluid dynamics will be revolutionized in the next decade as the power of Symbolic Manipulation becomes widely recognized.

12. FUTURE WORK. Besides the use of Symbolic Manipulation described above, we expect to extend the work described herein in the near future to include the following: unsteady equations, 3-dimensional problems, magnetic effects (which give rise to a tensor conductivity), dielectric interior boundaries (which require the precise control of the grid at interior points), solution adaptive methods to better resolve the maxima in the E-fields, and semi-automated optimization of the electrode design procedure.

ACKNOWLEDGEMENTS.

E. Saleski of LASL, J. Filcoff of AFWL and W. Moeny of Tetra Corporation provided orientation and references on the laser problems. H. Happ and D. Harrison of Tetra Corporation have provided programming support and code validation efforts. Prof. Stanly Steinberg of the Department of Mathematics and Statistics, university of New Mexico, is primarily responsible for the

Symbolic Manipulation work. The work described herein has been partially supported by the U.S. Army Research Office, the U.S. Air Force Weapons Laboratory, and the U.S. Air Force Office of Scientific Research.

REFERENCES.

1. Saleski, E. (personal communication).
2. Theophanis, G. A., Jacob, J. H. and Sackett, S. J. (1975), Jour. Applied Physics, 46, 2329.
3. Thompson, J. F., Thames, F. C. and Mastin, C. W. (1974), Jour. Computational Physics, 15, 299.
4. Roache, P. J. (1982), Semidirect/Marching Methods for Partial Differential Equations, to appear.
5. Roache, P. J. (1978), Numerical Heat Transfer. Part 1;1, i. Part 2;1, 163. Part 3;1, 183.
6. Roache, P. J. (1981), Numerical Heat Transfer, 4, 395.
7. Roache, P. J. (1978), Computers and Fluids, 3, 179.
8. Roache, P. J. (1981), Proc. Symposium on Numerical and Physical Aspects of Aerodynamic Flows, California State University at Long Beach, 19-21 January 1981.
9. Ghia, U. and Thames, F. C. (personal communications).
10. Maliska, C. R. (1981), A Solution Method for Three-Dimensional Fluid Flow Problems in Nonorthogonal Coordinates, Ph.D. dissertation, University of Waterloo.
11. Eiseman, P. R. (1979), Jour. Computational Physics, 33, 118.
12. Sorenson, R. L. (1980), A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation. NASA TM 81198, May 1980.
13. Steger, J. L., Sorenson, R. L. (1979), J. Comp. Phys., 33, 405.

1982 Army Numerical Analysis And Computer Conference and Tutorial

Attendees

Dale A. Anderson, Iowa State University
Roshdy S. Barsoum, AMMRC
Ammon Bierenzvige, Chemical System Lab.
Aivars Celmins, BRL
Jagdish Chandra, U.S. Army Research Office
Peter C. T. Chen, Benet Weapons Lab.
James Chew, North Carolina A&T
Terence P. Coffee, BRL
Herbert E. Cohen, AMSAA
Carl de Boer, MRC
Dale D. Ellis, Defence Research Establishment
John Erkman, Naval Surface Weapons Center
Joseph E. Flaherty, Benet Weapons Lab.
R. J. Gelinas, Science Applications, Inc.
Peter L. Green, US Army Missile Command
Frederick H. Gregory, BRL
Joseph M. Heimeril, BRL
Barry E. Herchenroder, CERC
Moayyed A. Hussain, General Electric Research Center
James M. Hyman, Los Alamos National Lab.
Billy Z. Jenkins, USA Missile Command
Robert L. Launer, U.S. Research Office
Luciano L. Leggio, ARRADCOM
C. Wayne Mastin, Mississippi State University
Gunter H. Meyer, Georgia Institute of Technology
Alfred H. Morris, Jr., Naval Surface Weapons Center
Charles J. Nietubice, BRL
Ben Nobel, MRC
John Nohel, MRC
Seymour V. Parter, MRC
Bradley Flohr, The Rockefeller University
Louis B. Rall, MRC
Patrick Roache, Ecodynamics Research Associates
Joseph M. Santiago, BRL
Brian R. Scott, BRL
C. N. Shen, Benet Weapons Lab.
Y. Peter Sheng, Aeronautical Research Associates
Joe F. Thompson, Mississippi State University
Royce W. Soanes, Benet Weapons Lab.
Devinder S. Sodhi, CRREL
Hyman M. Sternberg, Naval Surface Weapons Center
John Strikwerda, MRC
John D. Vasilakis, Benet Weapons Lab.
Julian J. Wu, Benet Weapons Lab.
Rao Yalamanchili, ARRADCOM

Attendees from the host installation are listed on the next page.

1982 Army Numerical Analysis And Computer Conference And Tutorial

WES Attendees

Byron J. Armstrong, Jr.
R. F. Athow
Donal Bach
R. C. Berger
Robert S. Bernard
Bill Boyt
H. Lee Butler
Ken Cargill
R. Case
Ray Chapman
J. Cheek
George W. Deer
Barbara P. Donnell
Mark Dortch
Paul Farrar
Mark D. Flohr
Jeff Earickson
Bruce Ebersole
Samuel B. Heltzel
Jeffery P. Holland
Jim Houston
Carl Huval
Yu-Shih Jeng
Billy Johnson
L. D. Johnson

Linda Johnson
Mark Johnson
Daniel Leavell
Allan S. Lessem
Joe Letter
Paul McCoy
Roger H. Multer
Frank Neilson
John F. Peters
Richard W. Peterson
Mark Prater
Norman Scheffner
Dick Schmalz
Paul Senter
Aaron Stein
Phillip Stewart
A. Swain
Allen Teeter
Barbara A. Tracy
Fred Tracy
S. Rao Vemulakonda
Ken P. Vitaya-Udom
Jack B. Waide
R. Weiss
David Williams

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM																										
1. REPORT NUMBER ARO Report 82-3	2. GOVT ACCESSION NO. AD-A118920	3. RECIPIENT'S CATALOG NUMBER																										
4. TITLE (and Subtitle) Proceedings of the 1982 Army Numerical Analysis and Computers Conference		5. TYPE OF REPORT & PERIOD COVERED Interim Technical Report																										
7. AUTHOR(s)		6. PERFORMING ORG. REPORT NUMBER																										
9. PERFORMING ORGANIZATION NAME AND ADDRESS		8. CONTRACT OR GRANT NUMBER(s)																										
11. CONTROLLING OFFICE NAME AND ADDRESS Army Mathematics Steering Committee on behalf of the Chief of Research, Development and Acquisition		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS																										
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) U. S. Army Research Office ATTN: DRXRO-MA P. O. Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE August 1982																										
		13. NUMBER OF PAGES 606																										
		15. SECURITY CLASS. (of this report)																										
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE																										
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.																												
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																												
18. SUPPLEMENTARY NOTES This is a technical report resulting from the 1982 Army Numerical Analysis and Computers Conference. It contains papers on computer aided designs and engineering as well as papers on numerical analysis.																												
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)																												
<table border="0"> <tr> <td>boundary-fitted coordinate code</td> <td>shallow-water equations</td> </tr> <tr> <td>flux-corrected transport</td> <td>asymptotic solutions</td> </tr> <tr> <td>grid generation techniques</td> <td>magnetic field coils</td> </tr> <tr> <td>hydrodynamic problems</td> <td>basin oscillation analysis</td> </tr> <tr> <td>random choice methods</td> <td>tsunami generation</td> </tr> <tr> <td>two-phase interior ballistics</td> <td>non-linear energy transfer</td> </tr> <tr> <td>free boundary problems</td> <td>stress wave problems</td> </tr> <tr> <td>autofrettaged tube</td> <td>adjoint variational methods</td> </tr> <tr> <td>finite element methods</td> <td>Stokes and Navier-Stokes equations</td> </tr> <tr> <td>heat conduction</td> <td>model of lake currents</td> </tr> <tr> <td>programming language</td> <td>adaptive grid methods</td> </tr> <tr> <td>lined gun barrels</td> <td>interactive designs using elliptic grids and semidirect/marching methods</td> </tr> <tr> <td>computer arithmetic</td> <td>mathematical software</td> </tr> </table>			boundary-fitted coordinate code	shallow-water equations	flux-corrected transport	asymptotic solutions	grid generation techniques	magnetic field coils	hydrodynamic problems	basin oscillation analysis	random choice methods	tsunami generation	two-phase interior ballistics	non-linear energy transfer	free boundary problems	stress wave problems	autofrettaged tube	adjoint variational methods	finite element methods	Stokes and Navier-Stokes equations	heat conduction	model of lake currents	programming language	adaptive grid methods	lined gun barrels	interactive designs using elliptic grids and semidirect/marching methods	computer arithmetic	mathematical software
boundary-fitted coordinate code	shallow-water equations																											
flux-corrected transport	asymptotic solutions																											
grid generation techniques	magnetic field coils																											
hydrodynamic problems	basin oscillation analysis																											
random choice methods	tsunami generation																											
two-phase interior ballistics	non-linear energy transfer																											
free boundary problems	stress wave problems																											
autofrettaged tube	adjoint variational methods																											
finite element methods	Stokes and Navier-Stokes equations																											
heat conduction	model of lake currents																											
programming language	adaptive grid methods																											
lined gun barrels	interactive designs using elliptic grids and semidirect/marching methods																											
computer arithmetic	mathematical software																											

